

Categories and all that — A Tutorial

Ernst-Erich Doberkat
Chair for Software Technology
Technische Universität Dortmund
`doberkat@acm.org`

April 1, 2014

Contents

1	Categories	2
1.1	Basic Definitions	3
1.2	Elementary Constructions	15
1.2.1	Products and Coproducts	15
1.2.2	Pullbacks and Pushouts	22
1.3	Functors and Natural Transformations	27
1.3.1	Functors	27
1.3.2	Natural Transformations	34
1.3.3	Limits and Colimits	40
1.4	Monads and Kleisli Triples	42
1.4.1	Kleisli Triples	43
1.4.2	Monads	44
1.4.3	Monads in <code>Haskell</code>	51
1.5	Adjunctions and Algebras	53
1.6	Coalgebras	67
1.6.1	Bisimulations	70
1.6.2	Congruences	83
1.7	Modal Logics	86
1.7.1	Frames and Models	89
1.7.2	The Lindenbaum Construction	101
1.7.3	Coalgebraic Logics	107
1.8	Bibliographic Notes	115
1.9	Exercises	116
	References	125
	Index	127

1 Categories

Many areas of Mathematics show surprising structural similarities, which suggests that it might be interesting and helpful to focus on an abstract view, unifying concepts. This abstract view looks at the mathematical objects from the outside and studies the relationship between them, for example groups (as objects) and homomorphisms (as an indicator of their relationship), or topological spaces together with continuous maps, or ordered sets with monotone maps, the list could be extended. It leads to the general notion of a category. A category is based on a class of objects together with morphisms for each two objects, which can be composed; composition follows some laws which are considered evident and natural.

This is an approach which has considerable appeal to a software engineer as well. In software engineering, the implementation details of a software system are usually not particularly important from an architectural point of view, they are encapsulated in a component. In contrast, the relationship of components to each other is of interest because this knowledge is necessary for composing a system from its components. Roughly speaking, the architecture of a software system is characterized both through its components and their interaction, the static part of which can be described through what we may perceive as morphisms.

This has been recognized fairly early in the software architecture community, witnessed by the April 1995 issue of the *IEEE Transactions on Software Engineering*, which was devoted to software architecture and introduced some categorical language in discussing architectures. So the language of categories offers some attractions to software engineers, as can be seen from, e.g., [Fia05, Bar01, Dob03]. We will also see that the tool set of modal logics, another area which is important to software construction, profits substantially from constructions which are firmly grounded in categories.

We will discuss categories here and introduce the reader to the basic constructions. The world of categories is too rich to be captured in these few pages, so we have made an attempt to provide the reader with some basic proficiency in categories, helping in learning her or him to get a grasp on the recent literature. This modest goal is attained by blending the abstract mathematical development with a plethora of examples. Exercises provide an opportunity to practice understanding and to give hints at developments not spelled out in the text.

We give a brief overview over the contents.

Overview. The definition of a category and a discussion of their most elementary properties is done in Section 1.1, examples show that categories are indeed a very versatile and general instrument for mathematical modelling. Section 1.2 discusses constructions like product and coproduct, which are familiar from other contexts, in this new language, and we look at pushouts and pullbacks, first in the familiar context of sets, then in a more general setting. Functors are introduced in Section 1.3 for relating categories to each other, and natural transformations permit functors to enter a relationship. We show also that set valued functors play a special rôle, which gives occasion to investigate more deeply the hom-sets of a category. Products and coproducts have an appearance again, but this time as instances of the more general concept of limits resp. colimits.

Monads and Kleisli tripel as very special functors are introduced and discussed in Section 1.4,

their relationship is investigated, and some examples are given, which provide an idea about the usefulness of this concept; a small section on monads in the programming language `Haskell` provides a pointer to the practical use of monads. Next we show that monads are generated from adjunctions. This important concept is introduced and discussed in 1.5; we define adjunctions, show by examples that adjunctions are a colorfully blooming and nourished flower in the garden of Mathematics, and give an alternative formulation in terms of units and counits; we then show that each adjunction gives us a monad, and that each monad also generates an adjunction. The latter part is interesting since it gives the occasion of introducing the algebras for a monad; we discuss two examples fairly extensively, indicating what such algebras might look like.

While an algebra provides a morphism $\mathbf{F}\mathbf{a} \rightarrow \mathbf{a}$, a coalgebra provides a morphism $\mathbf{a} \rightarrow \mathbf{F}\mathbf{a}$. This is introduced and discussed in Section 1.6, many examples show that this concept models a broad variety of applications in the area of systems. Coalgebras and their properties are studied, among them bisimulations, a concept which originates from the theory of concurrent systems and which is captured now coalgebraically. The Kripke models for modal logics provide an excellent play ground for coalgebras, so they are introduced in Section 1.7, examples show the broad applicability of this concept (but neighborhood models as a generalization are introduced as well). We go a bit beyond a mere application of coalgebras and give also the construction of the canonical model through Lindenbaum's construction of maximally consistent sets, which by the way provide an application of transfinite induction as well. We finally show that coalgebras may be put to use when constructing coalgebraic logics, a very fruitful and general approach to modal logics and their generalizations.

1.1 Basic Definitions

We will define what a category is, and give some examples for categories. It shows that this is a very general notion, covering also many formal structures that are studied in theoretical computer science. A very rough description would be to say that a category is a bunch of objects which are related to each other, the relationships being called morphisms. This gives already the gist of the definition — objects which are related to each other. But the relationship has to be made a bit more precise for being amenable to further investigation. So this is the definition of a category.

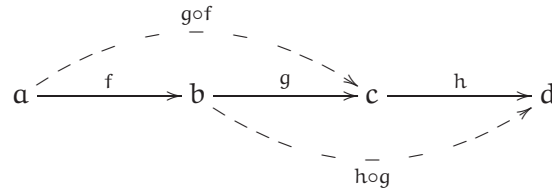
Definition 1.1 *A category \mathbf{K} consists of a class $|\mathbf{K}|$ of objects and for any objects \mathbf{a}, \mathbf{b} in $|\mathbf{K}|$ of a set $\text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{b})$ of morphisms with a composition operation \circ , mapping $\text{hom}_{\mathbf{K}}(\mathbf{b}, \mathbf{c}) \times \text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{b})$ to $\text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{c})$ with the following properties*

Identity *For every object \mathbf{a} in $|\mathbf{K}|$ there exists a morphism $\text{id}_{\mathbf{a}} \in \text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{a})$ with $f \circ \text{id}_{\mathbf{a}} = f = \text{id}_{\mathbf{b}} \circ f$, whenever $f \in \text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{b})$,*

Associativity *If $f \in \text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{b})$, $g \in \text{hom}_{\mathbf{K}}(\mathbf{b}, \mathbf{c})$ and $h \in \text{hom}_{\mathbf{K}}(\mathbf{c}, \mathbf{d})$, then $h \circ (g \circ f) = (h \circ g) \circ f$.*

Note that we do not think that a category is based on a set of objects (which would yield difficulties) but rather than on a class. In fact, if we would insist on having a set of objects, we could not talk about the category of sets, which is an important species for a category. We insist, however, on having *sets* of morphisms, because we want morphisms to be somewhat

clearly represented. Usually we write for $f \in \text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{b})$ also $f : \mathbf{a} \rightarrow \mathbf{b}$, if the context is clear. Thus if $f : \mathbf{a} \rightarrow \mathbf{b}$ and $g : \mathbf{b} \rightarrow \mathbf{c}$, then $g \circ f : \mathbf{a} \rightarrow \mathbf{c}$; one may think that first f is applied (or executed), and then g is applied to the result of f . Note the order in which the application is written down: $g \circ f$ means that g is applied to the result of f . The first postulate says that there is an *identity morphism* $\text{id}_{\mathbf{a}} : \mathbf{a} \rightarrow \mathbf{a}$ for each object \mathbf{a} of \mathbf{K} which does not have an effect on the other morphisms upon composition, so no matter if you do $\text{id}_{\mathbf{a}}$ first and then morphism $f : \mathbf{a} \rightarrow \mathbf{b}$, or if you do f first and then $\text{id}_{\mathbf{b}}$, you end up with the same result as if doing only f . Associativity is depicted through this diagram



Hence is you take the fast train $g \circ f$ from \mathbf{a} to \mathbf{c} first (no stop at \mathbf{b}) and then switch to train h or is you travel first with f from \mathbf{a} to \mathbf{b} and then change to the fast train $h \circ g$ (no stop at \mathbf{c}), you will end up with the same result.

Given $f \in \text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{b})$, we call object \mathbf{a} the *domain*, object \mathbf{b} the *codomain* of morphism f .

Let us have a look at some examples.

Example 1.2 The category **Set** is the most important of them all. It has sets as its class of objects, and the morphisms $\text{hom}_{\mathbf{Set}}(\mathbf{a}, \mathbf{b})$ are just the maps from set \mathbf{a} to set \mathbf{b} . The identity map $\text{id}_{\mathbf{a}} : \mathbf{a} \rightarrow \mathbf{a}$ maps each element to itself, and composition is just composition of maps, which is associative:

$$\begin{aligned} (f \circ (g \circ h))(x) &= f(g \circ h(x)) \\ &= f(g(h(x))) \\ &= (f \circ g)(h(x)) \\ &= ((f \circ g) \circ h)(x) \end{aligned}$$




The next example shows that one class of objects can carry more than one definition of morphisms.

Example 1.3 The category **Rel** has sets as its class of objects. Given sets \mathbf{a} and \mathbf{b} , $f \in \text{hom}_{\mathbf{Rel}}(\mathbf{a}, \mathbf{b})$ is a morphism from \mathbf{a} to \mathbf{b} iff $f \subseteq \mathbf{a} \times \mathbf{b}$ is a relation. Given set \mathbf{a} , define

$$\text{id}_{\mathbf{a}} := \{\langle x, x \rangle \mid x \in \mathbf{a}\}$$

as the identity relation, and define for $f \in \text{hom}_{\mathbf{Rel}}(\mathbf{a}, \mathbf{b})$, $g \in \text{hom}_{\mathbf{Rel}}(\mathbf{b}, \mathbf{c})$ the composition as

$$g \circ f := \{\langle x, z \rangle \mid \text{there exists } y \in \mathbf{b} \text{ with } \langle x, y \rangle \in f \text{ and } \langle y, z \rangle \in g\}$$

Because existential quantifiers can be interchanged, composition is associative, and $\text{id}_{\mathbf{a}}$ serves in fact as the identity element for composition. 

But morphisms do not need to be maps or relations.

Example 1.4 Let (P, \leq) be a partially ordered set. Define \mathbf{P} by taking the class $|\mathbf{P}|$ of objects as P , and put

$$\text{hom}_{\mathbf{P}}(p, q) := \begin{cases} \{\langle p, q \rangle\}, & \text{if } p \leq q \\ \emptyset, & \text{otherwise.} \end{cases}$$

Then id_p is $\langle p, p \rangle$, the only element of $\text{hom}_{\mathbf{P}}(p, p)$, and if $f : p \rightarrow q, g : q \rightarrow r$, thus $p \leq q$ and $q \leq r$, hence by transitivity $p \leq r$, so that we put $g \circ f := \langle p, r \rangle$. Let $h : r \rightarrow s$, then

$$\begin{aligned} h \circ (g \circ f) &= h \circ \langle p, r \rangle \\ &= \langle p, s \rangle \\ &= \langle q, s \rangle \circ f \\ &= (h \circ g) \circ f \end{aligned}$$

It is clear that $\text{id}_p = \langle p, p \rangle$ serves as a neutral element. \mathbb{M}

A directed graph generates a category through all its finite paths. Composition of two paths is then just their combination, indicating movement from one node to another, possibly via intermediate nodes. But we also have to cater for the situation that we want to stay in a node.

Example 1.5 Let $\mathcal{G} = (V, E)$ be a directed graph. Recall that a *path* $\langle p_0, \dots, p_n \rangle$ is a finite sequence of nodes such that adjacent nodes form an edge, i.e., such that $\langle p_i, p_{i+1} \rangle \in E$ for $0 \leq i < n$; each node a has an empty path $\langle a, a \rangle$ attached to it, which may or may not be an edge in the graph. The objects of the category $F(\mathcal{G})$ are the nodes V of \mathcal{G} , and a morphism $a \rightarrow b$ in $F(\mathcal{G})$ is a path connecting a with b in \mathcal{G} , hence a path $\langle p_0, \dots, p_n \rangle$ with $p_0 = a$ and $p_n = b$. The empty path serves as the identity morphism, the composition of morphism is just their concatenation; this is plainly associative. This category is called the *free category generated by graph \mathcal{G}* . \mathbb{M}

These two examples base categories on a set of objects; they are instances of small categories. A category is called *small* iff the objects form a set (rather than a class).

The discrete category is a trivial but helpful example.

Example 1.6 Let $X \neq \emptyset$ be a set, and define a category \mathbf{K} through $|\mathbf{K}| := X$ with

$$\text{hom}_{\mathbf{K}}(x, y) := \begin{cases} \text{id}_x, & x = y \\ \emptyset, & \text{otherwise} \end{cases}$$

This is the *discrete category* on X . \mathbb{M}

Algebraic structures furnish a rich and plentiful source for examples. Let us have a look at groups, and at Boolean algebras.

Example 1.7 The category of groups has as objects all groups (G, \cdot) , and as morphisms $f : (G, \cdot) \rightarrow (H, *)$ all maps $f : G \rightarrow H$ which are group homomorphisms, i.e., for which $f(1_G) = 1_H$ (with $1_G, 1_H$ as the respective neutral elements), for which $f(a^{-1}) = (f(a))^{-1}$ and $f(a \cdot b) = f(a) * f(b)$ always holds. The identity morphism $\text{id}_{(G, \cdot)}$ is the identity map, and

composition of homomorphisms is composition of maps. Because composition is inherited from category **Set**, we do not have to check for associativity or for identity.

Note that we did not give the category a particular name, it is simply referred to as the *category of groups*. 🙌

Example 1.8 Similarly, the category of Boolean algebras has Boolean algebras as objects, and a morphism $f : G \rightarrow H$ for the Boolean algebras G and H is a map f between the carrier sets with these properties:

$$\begin{aligned} f(-a) &= -f(a) \\ f(a \wedge b) &= f(a) \wedge f(b) \\ f(\top) &= \top \end{aligned}$$

(hence also $f(\perp) = \perp$, and $f(a \vee b) = f(a) \vee f(b)$). Again, composition of morphisms is composition of maps, and the identity morphism is just the identity map. 🙌

The next example deals with transition systems. Formally, a transition system is a directed graph. But whereas discussing a graph puts the emphasis usually on its paths, a transition system is concerned more with the study of, well, the transition from one state to another one, hence the focus is usually stronger localized. This is reflected also when defining morphisms, which, as we will see, come in two flavors.

Example 1.9 A *transition system* (S, \rightsquigarrow_S) is a set S of states together with a transition relation $\rightsquigarrow_S \subseteq S \times S$. Intuitively, $s \rightsquigarrow_S s'$ iff there is a transition from s to s' . Transition systems form a category: the objects are transition systems, and a morphism $f : (S, \rightsquigarrow_S) \rightarrow (T, \rightsquigarrow_T)$ is a map $f : S \rightarrow T$ such that $s \rightsquigarrow_S s'$ implies $f(s) \rightsquigarrow_T f(s')$. This means that a transition from s to s' in (S, \rightsquigarrow_S) entails a transition from $f(s)$ to $f(s')$ in the transition system (T, \rightsquigarrow_T) . Note that the defining condition for f can be written as $\rightsquigarrow_S \subseteq (f \times f)^{-1}[\rightsquigarrow_T]$ with $f \times f : \langle s, s' \rangle \mapsto \langle f(s), f(s') \rangle$. 🙌

The morphisms in Example 1.9 are interesting from a relational point of view. We will require an additional property which, roughly speaking, makes sure that we not only transport transitions through morphisms, but that we are also able to capture transitions which emanate from the image of a state. So we want to be sure that, if $f(s) \rightsquigarrow_T t$, we obtain this transition from a transition arising from s in the original system. This idea is formulated in the next example, it will arise again in a very natural manner in Example 1.135 in the context of coalgebras.


Example 1.10 We continue with transition systems, so we define a category which has transition systems as objects. A *morphism* $f : (S, \rightsquigarrow_S) \rightarrow (T, \rightsquigarrow_T)$ in the present category is a map $f : S \rightarrow T$ such that for all $s, s' \in S, t \in T$

Forward: $s \rightsquigarrow_S s'$ implies $f(s) \rightsquigarrow_T f(s')$,

Backward: if $f(s) \rightsquigarrow_T t'$, then there exists $s' \in S$ with $f(s') = t'$ and $s \rightsquigarrow_S s'$.

The forward condition is already known from Example 1.9, the backward condition is new. It states that if we start a transition from some $f(s)$ in T , then this transition originates from some transition starting from s in S ; to distinguish these morphisms from the ones considered in Example 1.9, they are called *bounded* morphisms. The identity map $S \rightarrow S$ yields a bounded


morphism, and the composition of bounded morphisms is a bounded morphism again. In fact, let $f : (S, \rightsquigarrow_S) \rightarrow (T, \rightsquigarrow_T)$, $g : (T, \rightsquigarrow_T) \rightarrow (U, \rightsquigarrow_U)$ be bounded morphisms, and assume that $g(f(s)) \rightsquigarrow_U u'$. Then we can find $t' \in T$ with $g(t') = u'$ and $f(s) \rightsquigarrow_T t'$, hence we find $s' \in S$ with $f(s') = t'$ and $s \rightsquigarrow_S s'$.

Bounded morphisms are of interest in the study of models for modal logics [BdRV01], see Lemma 1.193. 


The next examples reverse arrows when it comes to define morphisms. The examples so far observed the effects of maps in the direction in which the maps were defined. We will, however, also have an opportunity to look back, and to see what properties the inverse image of a map is supposed to have. We study this in the context of measurable, and of topological spaces.

Example 1.11 Let S be a set, and assume that \mathcal{A} is a σ -algebra on S . Then the pair (S, \mathcal{A}) is called a *measurable space*, the elements of the σ -algebra are sometimes called \mathcal{A} -measurable sets. The category **Meas** has as objects all measurable spaces.

Given two measurable spaces (S, \mathcal{A}) and (T, \mathcal{B}) , a map $f : S \rightarrow T$ is called a *morphism of measurable spaces* iff f is \mathcal{A} - \mathcal{B} -measurable. This means that $f^{-1}[B] \in \mathcal{A}$ for all $B \in \mathcal{B}$, hence the set $\{s \in S \mid f(s) \in B\}$ is an \mathcal{A} -measurable set for each \mathcal{B} -measurable set B . Each σ -algebra is a Boolean algebra, but the definition of a morphism of measurable spaces does not entail that such a morphism induces a morphism of Boolean algebras (see Example 1.8). Consequently the behavior of f^{-1} rather than the one of f determines whether f belongs to the distinguished set of morphisms.

Thus the \mathcal{A} - \mathcal{B} -measurable maps $f : S \rightarrow T$ are the morphisms $f : (S, \mathcal{A}) \rightarrow (T, \mathcal{B})$ in category **Meas**. The identity morphism on (S, \mathcal{A}) is the identity map (this map is measurable because $\text{id}^{-1}[A] = A \in \mathcal{A}$ for each $A \in \mathcal{A}$). Composition of measurable maps yields a measurable map again: let $f : (S, \mathcal{A}) \rightarrow (T, \mathcal{B})$ and $g : (T, \mathcal{B}) \rightarrow (U, \mathcal{C})$, then $(g \circ f)^{-1}[D] = f^{-1}[g^{-1}[D]] \in \mathcal{A}$, for $D \in \mathcal{C}$, because $g^{-1}[D] \in \mathcal{B}$. It is clear that composition is associative, since it is based on composition of ordinary maps. 

The next example deals with topologies, which are of course also sets of subsets. Continuity is formulated similar to measurably in terms of the inverse rather than the direct image.

Example 1.12 Let S be a set and \mathcal{G} be a topology on S ; hence $\mathcal{G} \subseteq \mathcal{P}(S)$ such that $\emptyset, S \in \mathcal{G}$, \mathcal{G} is closed under finite intersections and arbitrary unions. Then (S, \mathcal{G}) is called a *topological space*. Given another topological space (T, \mathcal{H}) , a map $f : S \rightarrow T$ is called \mathcal{G} - \mathcal{H} -continuous iff the inverse image of an open set is open again, i.e., iff $f^{-1}[G] \in \mathcal{G}$ for all $G \in \mathcal{H}$. Category **Top** of topological spaces has all topological spaces as objects, and continuous maps as morphisms. The identity $(S, \mathcal{G}) \rightarrow (S, \mathcal{G})$ is certainly continuous. Again, it follows that the composition of morphisms yields a morphism, and that their composition is associative. 

Now that we know what a category is, we start constructing new categories from given ones. We begin by building on category **Meas** another interesting category, indicating that a category can be used as a building block for another one.

Example 1.13 A measurable space (S, \mathcal{A}) together with a probability measure μ on \mathcal{A} is called a *probability space* and written as (S, \mathcal{A}, μ) . The category **Prob** of all probability spaces has — you guessed it — as objects all probability spaces; a morphism $f : (S, \mathcal{A}, \mu) \rightarrow (T, \mathcal{B}, \nu)$

is a morphism $f : (S, \mathcal{A}) \rightarrow (T, \mathcal{B})$ in **Meas** for the underlying measurable spaces such that $\nu(B) = \mu(f^{-1}[B])$ holds for all $B \in \mathcal{B}$. Thus the ν -probability for event $B \in \mathcal{B}$ is the same as the μ -probability for all those $s \in S$ the image of which is in B . Note that $f^{-1}[B] \in \mathcal{A}$ due to f being a morphism in **Meas**, so that $\mu(f^{-1}[B])$ is in fact defined. \mathfrak{M}

We go a bit further and combine two measurable spaces into a third one; this requires adjusting the notion of a morphism, which are in this new category basically pairs of morphisms from the underlying category. This shows the flexibility with which we may — and do — manipulate morphisms.

Example 1.14 Denote for the measurable space (S, \mathcal{A}) by $\mathbb{S}(S, \mathcal{A})$ the set of all subprobability measures. Define

$$\begin{aligned}\beta_S(A, r) &:= \{\mu \in \mathbb{S}(S, \mathcal{A}) \mid \mu(A) \geq r\}, \\ w(\mathcal{A}) &:= \sigma(\{\beta_S(A, r) \mid A \in \mathcal{A}, 0 \leq r \leq 1\})\end{aligned}$$

Thus $\beta_S(A, r)$ denotes all probability measures which evaluate the set A not smaller than r , and $w(\mathcal{A})$ collects all these sets into a σ -algebra; w alludes to “weak”, $w(\mathcal{A})$ is sometimes called the *weak- σ -algebra* or the *weak σ -algebra* associated with \mathcal{A} as the σ -algebra generated by the family of sets. This makes $(\mathbb{S}(S, \mathcal{A}), w(\mathcal{A}))$ a measurable space, based on the probabilities over (S, \mathcal{A}) .

Let (T, \mathcal{B}) be another measurable space. A map $K : S \rightarrow \mathbb{S}(T, \mathcal{B})$ is \mathcal{A} - $w(\mathcal{B})$ -measurable iff $\{s \in S \mid K(s)(B) \geq r\} \in \mathcal{A}$ for all $B \in \mathcal{B}$; this follows from Exercise 7. We take as objects for our category the triplets $((S, \mathcal{A}), (T, \mathcal{B}), K)$, where (S, \mathcal{A}) and (T, \mathcal{B}) are measurable spaces and $K : S \rightarrow \mathbb{S}(T, \mathcal{B})$ is \mathcal{A} - $w(\mathcal{B})$ -measurable. A morphism $(f, g) : ((S, \mathcal{A}), (T, \mathcal{B}), K) \rightarrow ((S', \mathcal{A}'), (T', \mathcal{B}'), K')$ is a pair of morphisms $f : (S, \mathcal{A}) \rightarrow (S', \mathcal{A}')$ and $g : (T, \mathcal{B}) \rightarrow (T', \mathcal{B}')$ such that

$$K(s)(g^{-1}[B']) = K'(f(s))(B')$$

holds for all $s \in S$ and for all $B' \in \mathcal{B}'$.

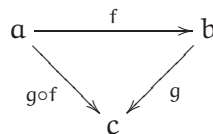
The composition of morphisms is defined component wise:

$$(f', g') \circ (f, g) := (f' \circ f, g' \circ g).$$

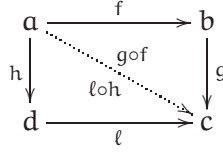
Note that $f' \circ f$ and $g' \circ g$ refer to the composition of maps, while $(f', g') \circ (f, g)$ refers to the newly defined composition in our new spic-and-span category (we should probably use another symbol, but no confusion can arise, since the new composition operates on pairs). The identity morphism for $((S, \mathcal{A}), (T, \mathcal{B}), K)$ is just the pair $(\text{id}_S, \text{id}_T)$. Because the composition of maps is associative, composition in our new category is associative as well, and because $(\text{id}_S, \text{id}_T)$ is composed from identities, it is also an identity.

This category is sometimes called the *category of stochastic relations*. \mathfrak{M}

Before continuing, we introduce commutative diagrams. Suppose that we have in a category **K** morphisms $f : a \rightarrow b$ and $g : b \rightarrow c$. The combined morphism $g \circ f$ is represented graphically as

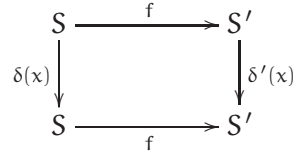


If the morphisms $h : a \rightarrow d$ and $\ell : d \rightarrow c$ satisfy $g \circ f = \ell \circ h$, we have a *commutative diagram*; in this case we do not draw out the morphism in the diagonal.



We consider automata next, to get some feeling for the handling of commutative diagrams, and as an illustration for an important formalism looked at through the glasses of categories.

Example 1.15 Given sets X and S of inputs and states, respectively, an *automaton* (X, S, δ) is defined by a map $\delta : X \times S \rightarrow S$. The interpretation is that $\delta(x, s)$ is the new state after input $x \in X$ in state $s \in S$. Reformulating, $\delta(x) : s \mapsto \delta(x, s)$ is perceived as a map $S \rightarrow S$ for each $x \in X$, so that the new state now is written as $\delta(x)(s)$; manipulating a map with two arguments in this way is called *currying* and will be considered in greater detail in Example 1.103. The objects of our category of automata are the automata, and an *automaton morphism* $f : (X, S, \delta) \rightarrow (X, S', \delta')$ is a map $f : S \rightarrow S'$ such that this diagram commutes for all $x \in X$:

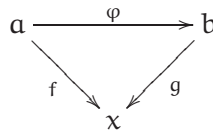


Hence we have $f(\delta(x)(s)) = \delta'(x)(f(s))$ for each $x \in X$ and $s \in S$; this means that computing the new state and mapping it through f yields the same result as computing the new state for the mapped one. The identity map $S \rightarrow S$ yields a morphism, hence automata form a category.

Note that morphisms are defined only for automata with the same input alphabet. This reflects the observation that the input alphabet is usually given by the environment, while the set of states represents a model about the automata's behavior, hence is at our disposal for manipulation. 🙌

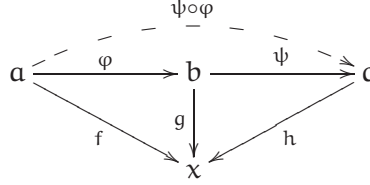
Whereas we constructed above new categories from given one in an ad hoc manner, categories also yield new categories systematically. This is a simple example.

Example 1.16 Let \mathbf{K} be a category; fix an object x on \mathbf{K} . The objects of our new category are the morphisms $f \in \text{hom}_{\mathbf{K}}(a, x)$ for an object a . Given objects $f \in \text{hom}_{\mathbf{K}}(a, x)$ and $g \in \text{hom}_{\mathbf{K}}(b, x)$ in the new category, a morphism $\varphi : f \rightarrow g$ is a morphism $\varphi \in \text{hom}_{\mathbf{K}}(a, b)$ with $f = g \circ \varphi$, so that this diagram commutes



Composition is inherited from \mathbf{K} . The identity $\text{id}_f : f \rightarrow f$ is $\text{id}_a \in \text{hom}_{\mathbf{K}}(a, a)$, provided $f \in \text{hom}_{\mathbf{K}}(a, x)$. Since the composition in \mathbf{K} is associative, we have only to make sure that

the composition of two morphisms is a morphism again. This can be read off the following diagram: $(\varphi \circ \psi) \circ h = \varphi \circ (\psi \circ h) = \varphi \circ g = f$.



This category is sometimes called the *slice category* \mathbf{K}/x ; the object x is interpreted as an index, so that a morphism $f : a \rightarrow x$ serves as an indexing function. A morphism $\varphi : a \rightarrow b$ in \mathbf{K}/x is then compatible with the index operation. 🙌

The next example reverses arrows while at the same time maintaining the same class of objects.

Example 1.17 Let \mathbf{K} be a category. We define \mathbf{K}^{op} , the category *dual* to \mathbf{K} , in the following way: the objects are the same as for the original category, hence $|\mathbf{K}^{\text{op}}| = |\mathbf{K}|$, and the arrows are reversed, hence we put $\text{hom}_{\mathbf{K}^{\text{op}}}(a, b) := \text{hom}_{\mathbf{K}}(b, a)$ for the objects a, b ; the identity remains the same. We have to define composition in this new category. Let $f \in \text{hom}_{\mathbf{K}^{\text{op}}}(a, b)$ and $g \in \text{hom}_{\mathbf{K}^{\text{op}}}(b, c)$, then $g * f := f \circ g \in \text{hom}_{\mathbf{K}^{\text{op}}}(a, c)$. It is readily verified that $*$ satisfies all the laws for composition from Definition 1.1.

The dual category is sometimes helpful because it permits to cast notions into a uniform framework. 🙌

Example 1.18 Let us look at \mathbf{Rel} again. The morphisms $\text{hom}_{\mathbf{Rel}^{\text{op}}}(S, T)$ from S to T in \mathbf{Rel}^{op} are just the morphisms $\text{hom}_{\mathbf{Rel}}(T, S)$ in \mathbf{Rel} . Take $f \in \text{hom}_{\mathbf{Rel}^{\text{op}}}(S, T)$, then $f \subseteq T \times S$, hence $f^t \subseteq S \times T$, where relation

$$f^t := \{\langle s, t \rangle \mid \langle t, s \rangle \in f\}$$

is the transposed of relation f . The map $f \mapsto f^t$ is injective and compatible with composition, moreover it maps $\text{hom}_{\mathbf{Rel}^{\text{op}}}(S, T)$ onto $\text{hom}_{\mathbf{Rel}}(T, S)$. But this means that \mathbf{Rel}^{op} is essentially the same as \mathbf{Rel} . 🙌

It is sometimes helpful to combine two categories into a product:

Lemma 1.19 *Given categories \mathbf{K} and \mathbf{L} , define the objects of $\mathbf{K} \times \mathbf{L}$ as pairs $\langle a, b \rangle$, where a is an object in \mathbf{K} , and b is an object in \mathbf{L} . A morphism $\langle a, b \rangle \rightarrow \langle a', b' \rangle$ in $\mathbf{K} \times \mathbf{L}$ is comprised of morphisms $a \rightarrow a'$ in \mathbf{K} and $b \rightarrow b'$ in \mathbf{L} . Then $\mathbf{K} \times \mathbf{L}$ is a category. \dashv*

We have a closer look at morphisms now. Experience tells us that injective and surjective maps are fairly helpful, so a characterization in a category might be desirable. There is a small but not insignificant catch, however. We have seen that morphisms are not always maps, so that we are forced to find a characterization purely in terms of composition and equality, because this is all we have in a category. The following characterization of injective maps provides a clue for a more general definition.

Proposition 1.20 *Let $f : X \rightarrow Y$ be a map, then these statements are equivalent.*

1. f is injective.

2. If A is an arbitrary set, $g_1, g_2 : A \rightarrow X$ are maps with $f \circ g_1 = f \circ g_2$, then $g_1 = g_2$

Proof $1 \Rightarrow 2$: Assume f is injective and $f \circ g_1 = f \circ g_2$, but $g_1 \neq g_2$. Thus there exists $x \in A$ with $g_1(x) \neq g_2(x)$. But $f(g_1(x)) = f(g_2(x))$, and since f is injective, $g_1(x) = g_2(x)$. This is a contradiction.

$2 \Rightarrow 1$: Assume the condition holds, but f is not injective. Then there exists $x_1 \neq x_2$ with $f(x_1) = f(x_2)$. Let $A := \{\star\}$ and put $g_1(\star) := x_1$, $g_2(\star) := x_2$, thus $f(x_1) = (f \circ g_1)(\star) = (f \circ g_2)(\star) = f(x_2)$. By the condition $g_1 = g_2$, thus $x_1 = x_2$. Another contradiction. \dashv

This leads to a definition of the category version of injectivity as a morphism which is cancellable on the left.

Definition 1.21 Let \mathbf{K} be a category, a, b objects in \mathbf{K} . Then $f : a \rightarrow b$ is called a monomorphism (or a monic) iff whenever $g_1, g_2 : x \rightarrow a$ are morphisms with $f \circ g_1 = f \circ g_2$, then $g_1 = g_2$.

These are some simple properties of monomorphisms, which are also sometimes called monos.

Lemma 1.22 In a category \mathbf{K} ,

1. The identity is a monomorphism.
2. The composition of two monomorphisms is a monomorphism again.
3. If $k \circ f$ is a monomorphism for some morphism k , then f is a monomorphism.

Proof The first part is trivial. Let $f : a \rightarrow b$ and $g : b \rightarrow c$ both monos. Assume $h_1, h_2 : x \rightarrow a$ with $h_1 \circ (g \circ f) = h_2 \circ (g \circ f)$. We want to show $h_1 = h_2$. By associativity $(h_1 \circ g) \circ f = (h_2 \circ g) \circ f$. Because f is a mono, we conclude $h_1 \circ g = h_2 \circ g$, because g is a mono, we see $h_1 = h_2$.

Finally, let $f : a \rightarrow b$ and $k : b \rightarrow c$. Assume $h_1, h_2 : x \rightarrow a$ with $f \circ h_1 = f \circ h_2$. We claim $h_1 = h_2$. Now $f \circ h_1 = f \circ h_2$ implies $k \circ f \circ h_1 = k \circ f \circ h_2$. Thus $h_1 = h_2$. \dashv

In the same way we characterize surjectivity purely in terms of composition (exhibiting a nice symmetry between the two notions).

Proposition 1.23 Let $f : X \rightarrow Y$ be a map, then these statements are equivalent.

1. f is surjective.
2. If B is an arbitrary set, $g_1, g_2 : Y \rightarrow B$ are maps with $g_1 \circ f = g_2 \circ f$, then $g_1 = g_2$

Proof $1 \Rightarrow 2$: Assume f is surjective, $g_1 \circ f = g_2 \circ f$, but $g_1(y) \neq g_2(y)$ for some y . If we can find $x \in X$ with $f(x) = y$, then $g_1(y) = (g_1 \circ f)(x) = (g_2 \circ f)(x) = g_2(y)$, which would be a contradiction. Thus $y \notin f[X]$, hence f is not onto.

$2 \Rightarrow 1$: Assume that there exists $y \in Y$ with $y \notin f[X]$. Define $g_1, g_2 : Y \rightarrow \{0, 1, 2\}$ through

$$g_1(y) := \begin{cases} 0, & \text{if } y \in f[X], \\ 1, & \text{otherwise.} \end{cases} \quad g_2(y) := \begin{cases} 0, & \text{if } y \in f[X], \\ 2, & \text{otherwise.} \end{cases}$$

Then $g_1 \circ f = g_2 \circ f$, but $g_1 \neq g_2$. This is a contradiction. \neg

This suggests a definition of surjectivity through a morphism which is right cancellable.

Definition 1.24 Let \mathbf{K} be a category, a, b objects in \mathbf{K} . Then $f : a \rightarrow b$ is called a *epimorphism* (or an *epic*) iff whenever $g_1, g_2 : b \rightarrow c$ are morphisms with $g_1 \circ f = g_2 \circ f$, then $g_1 = g_2$.

These are some important properties of epimorphisms, which are sometimes called epis:

Lemma 1.25 In a category \mathbf{K} ,

1. The identity is an epimorphism.
2. The composition of two epimorphisms is an epimorphism again.
3. If $f \circ k$ is an epimorphism for some morphism k , then f is an epimorphism.

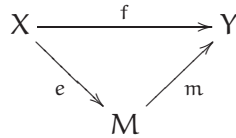
Proof We sketch the proof only for the the third part:

$$g_1 \circ f = g_2 \circ f \Rightarrow g_1 \circ f \circ k = g_2 \circ f \circ k \Rightarrow g_1 = g_2.$$

\neg

This is a small application of the decomposition of a map into an epimorphism and a monomorphism.

Proposition 1.26 Let $f : X \rightarrow Y$ be a map. Then there exists a factorization of f into $m \circ e$ with e an epimorphism and m a monomorphism.



The idea of the proof may best be described in terms of X as inputs, Y as outputs of system f . We collect all inputs with the same functionality, and assign each collection the functionality through which it is defined.

Proof Define

$$\ker(f) := \{\langle x_1, x_2 \rangle \mid f(x_1) = f(x_2)\}$$

(the *kernel* of f). This is an equivalence relation on X (*reflexivity*: $\langle x, x \rangle \in \ker(f)$ for all x , *symmetry*: if $\langle x_1, x_2 \rangle \in \ker(f)$ then $\langle x_2, x_1 \rangle \in \ker(f)$; *transitivity*: $\langle x_1, x_2 \rangle \in \ker(f)$ and $\langle x_2, x_3 \rangle \in \ker(f)$ together imply $\langle x_1, x_3 \rangle \in \ker(f)$).

Define

$$e : \begin{cases} X & \rightarrow X/\ker(f), \\ x & \mapsto [x]_{\ker(f)} \end{cases}$$

then e is an epimorphism. In fact, if $g_1 \circ e = g_2 \circ e$ for $g_1, g_2 : X/\ker(f) \rightarrow B$ for some set B , then $g_1(t) = g_2(t)$ for all $t \in X/\ker(f)$, hence $g_1 = g_2$.

Moreover

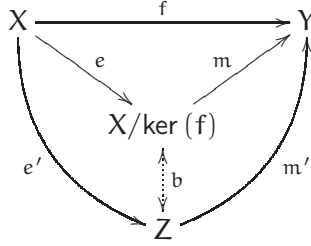
$$m : \begin{cases} X/\ker(f) & \rightarrow Y \\ [x]_{\ker(f)} & \mapsto f(x) \end{cases}$$

is well defined, since if $[x]_{\ker(f)} = [x']_{\ker(f)}$, then $f(x) = f(x')$, and m is a monomorphism. In fact, if $m \circ g_1 = m \circ g_2$ for arbitrary $g_1, g_2 : A \rightarrow X/\ker(f)$ for some set A , then $f(g_1(a)) = f(g_2(a))$ for all a , hence $\langle g_1(a), g_2(a) \rangle \in \ker(f)$. But this means $[g_1(a)]_{\ker(f)} = [g_2(a)]_{\ker(f)}$ for all $a \in A$, so $g_1 = g_2$. Evidently $f = m \circ e$. \dashv

Looking a bit harder at the diagram, we find that we can say even more, viz., that the decomposition is unique up to isomorphism.

Corollary 1.27 *If the map $f : X \rightarrow Y$ can be written as $f = e \circ m = e' \circ m'$ with epimorphisms e, e' and monomorphisms m, m' , then there is a bijection b with $e' = b \circ e$ and $m = m' \circ b$.*

Proof Since the composition of bijections is a bijection again, we may and do assume without loss of generality that $e : X \rightarrow X/\ker(f)$ maps x to its class $[x]_{\ker(f)}$, and that $m : X/\ker(f) \rightarrow Y$ maps $[x]_{\ker(f)}$ to $f(x)$. Then we have this diagram for the primed factorization $e' : X \rightarrow Z$ and $m' : Z \rightarrow Y$:



Note that

$$\begin{aligned} [x]_{\ker(f)} \neq [x']_{\ker(f)} &\Leftrightarrow f(x) \neq f(x') \\ &\Leftrightarrow m'(e'(x)) \neq m'(e'(x')) \\ &\Leftrightarrow e(x) \neq e(x') \end{aligned}$$

Thus defining $b([x]_{\ker(f)}) := e'(x)$ gives an injective map $X/\ker(f) \rightarrow Z$. Given $z \in Z$, there exists $x \in X$ with $e'(x) = z$, hence $b([x]_{\ker(f)}) = z$, thus b is onto. Finally, $m'(b([x]_{\ker(f)})) = m'(e'(x)) = f(x) = m([x]_{\ker(f)})$. \dashv

This factorization of a morphism is called an *epi/mono factorization*, and we just have shown that such a factorization is unique up to isomorphisms (a.k.a. bijections in **Set**).

The following example shows that epimorphisms are not necessarily surjective, even if they are maps.

Example 1.28 Recall that $(M, *)$ is a *monoid* iff $* : M \times M \rightarrow M$ is associative with a neutral element 0_M . For example, $(\mathbb{Z}, +)$ and (\mathbb{N}, \cdot) are monoids, so is the set X^* of all strings over alphabet X with concatenation as composition (with the empty string as neutral element). A *morphism* $f : (M, *) \rightarrow (N, \ddagger)$ is a map $f : M \rightarrow N$ such that $f(a * b) = f(a) \ddagger f(b)$, and $f(0_M) = 0_N$.

Now let $f : (\mathbb{Z}, +) \rightarrow (\mathbb{N}, \ddagger)$ be a morphism, then f is uniquely determined by the value $f(1)$. This is so since $m = 1 + \dots + 1$ (m times) for $m > 0$, thus $f(m) = f(1 + \dots + 1) = f(1) \ddagger \dots \ddagger f(1)$.

Also $f(-1) \dagger f(1) = f(-1 + 1) = f(0)$, so $f(-1)$ is inverse to $f(1)$, hence $f(-m)$ is inverse to $f(m)$. Consequently, if two morphisms map 1 to the same value, then the morphisms are identical.

Note that the inclusion $i : x \mapsto x$ is a morphism $i : (\mathbb{N}_0, +) \rightarrow (\mathbb{Z}, +)$. We claim that i is an epimorphism. Let $g_1 \circ i = g_2 \circ i$ for some morphisms $g_1, g_2 : (\mathbb{Z}, +) \rightarrow (M, *)$. Then $g_1(1) = (g_1 \circ i)(1) = (g_2 \circ i)(1) = g_2(1)$. Hence $g_1 = g_2$. Thus epimorphisms are not necessarily surjective. \upharpoonright

Composition induces maps between the hom sets of a category, which we are going to study now. Specifically, let \mathbf{K} be a fixed category, take objects a and b and fix for the moment a morphism $f : a \rightarrow b$. Then $g \mapsto f \circ g$ maps $\text{hom}_{\mathbf{K}}(x, a)$ to $\text{hom}_{\mathbf{K}}(x, b)$, and $h \mapsto h \circ f$ maps $\text{hom}_{\mathbf{K}}(b, x)$ to $\text{hom}_{\mathbf{K}}(a, x)$ for each object x . We investigate $g \mapsto f \circ g$ first. Define for an object x of \mathbf{K} the map

$$\text{hom}_{\mathbf{K}}(x, f) : \begin{cases} \text{hom}_{\mathbf{K}}(x, a) & \rightarrow \text{hom}_{\mathbf{K}}(x, b) \\ g & \mapsto f \circ g \end{cases}$$

Then $\text{hom}_{\mathbf{K}}(x, f)$ defines a map between morphisms, and we can determine through this map whether or not f is a monomorphism.

Lemma 1.29 $f : a \rightarrow b$ is a monomorphism iff $\text{hom}_{\mathbf{K}}(x, f)$ is injective for all objects x .

Proof This follows immediately from the observation

$$f \circ g_1 = f \circ g_2 \Leftrightarrow \text{hom}_{\mathbf{K}}(x, f)(g_1) = \text{hom}_{\mathbf{K}}(x, f)(g_2).$$

\dashv

Dually, define for an object x of \mathbf{K} the map

$$\text{hom}_{\mathbf{K}}(f, x) : \begin{cases} \text{hom}_{\mathbf{K}}(b, x) & \rightarrow \text{hom}_{\mathbf{K}}(a, x) \\ g & \mapsto g \circ f \end{cases}$$

Note that we change directions here: $f : a \rightarrow b$ corresponds to $\text{hom}_{\mathbf{K}}(b, x) \rightarrow \text{hom}_{\mathbf{K}}(a, x)$. Note also that we did reuse the name $\text{hom}_{\mathbf{K}}(\cdot)$; but no confusion should arise, because the signature tells us which map we specifically have in mind. Lemma 1.29 seems to suggest that surjectivity of $\text{hom}_{\mathbf{K}}(f, x)$ and f being an epimorphism are related. This, however, is not the case. But try this:

Lemma 1.30 $f : a \rightarrow b$ is an epimorphism iff $\text{hom}_{\mathbf{K}}(f, x)$ is injective for each object x .

Proof $\text{hom}_{\mathbf{K}}(f, x)(g_1) = \text{hom}_{\mathbf{K}}(f, x)(g_2)$ is equivalent to $g_1 \circ f = g_2 \circ f$. \dashv

Not surprisingly, an isomorphism is an invertible morphism; this is described in our scenario as follows.

Definition 1.31 $f : a \rightarrow b$ is called an isomorphism iff there exists a morphism $g : b \rightarrow a$ such that $g \circ f = \text{id}_a$ and $f \circ g = \text{id}_b$.

It is clear that morphism g is in this case uniquely determined: let g and g' be morphisms with the property above, then we obtain $g = g \circ \text{id}_b = g \circ (f \circ g') = (g \circ f) \circ g' = \text{id}_a \circ g' = g'$.

When we are in the category **Set** of sets with maps, an isomorphism f is bijective. In fact, let g be chosen to f according to Definition 1.31, then

$$\begin{aligned} h_1 \circ f &= h_2 \circ f \Rightarrow h_1 \circ f \circ g = h_2 \circ f \circ g \Rightarrow h_1 = h_2, \\ f \circ g_1 &= f \circ g_2 \Rightarrow g \circ f \circ g_1 = g \circ f \circ g_2 \Rightarrow g_1 = g_2, \end{aligned}$$

so that the first line makes f an epimorphism, and the second one a monomorphism.

The following lemma is often helpful (and serves as an example of the popular art of *diagram chasing*).

Lemma 1.32 *Assume that in this diagram*

$$\begin{array}{ccccc} a & \xrightarrow{f} & b & \xrightarrow{g} & c \\ \downarrow k & & \downarrow \ell & & \downarrow m \\ x & \xrightarrow{r} & y & \xrightarrow{s} & z \end{array}$$

the outer diagram commutes, that the leftmost diagram commutes, and that f is an epimorphism. Then the rightmost diagram commutes as well.

Proof In order to show that $m \circ g = s \circ \ell$ it is enough to show that $m \circ g \circ f = s \circ \ell \circ f$, because we then can cancel f , since f is an epi. But now

$$\begin{aligned} (m \circ g) \circ f &= m \circ (g \circ f) \\ &= (s \circ r) \circ k && \text{(commutativity of the outer diagram)} \\ &= s \circ (r \circ k) \\ &= s \circ (\ell \circ f) && \text{(commutativity of the leftmost diagram)} \\ &= (s \circ \ell) \circ f \end{aligned}$$

Now cancel f . \dashv

1.2 Elementary Constructions

In this section we deal with some elementary constructions, showing mainly how some important constructions for sets can be carried over to categories, hence are available in more general structures. Specifically, we will study products and sums (coproducts) as well as pullbacks and pushouts. We will not study more general constructs at present, in particular we will not have a look at limits and colimits. Once products and pullbacks are understood, the step to limits should not be too complicated, similarly for colimits, as the reader can see in the brief discussion in Section 1.3.3.

We fix a category **K**.

1.2.1 Products and Coproducts

The Cartesian product of two sets is just the set of pairs. In a general category we do not have a characterization through sets and their elements at our disposal, so we have to fill

this gap by going back to morphisms. Thus we require a characterization of product through morphisms. The first thought is using the projections $\langle x, y \rangle \mapsto x$ and $\langle x, y \rangle \mapsto y$, since a pair can be reconstructed through its projections. But this is not specific enough. An additional characterization of the projections is obtained through factoring: if there is another pair of maps pretending to be projections, they better be related to the “genuine” projections. This is what the next definition expresses.

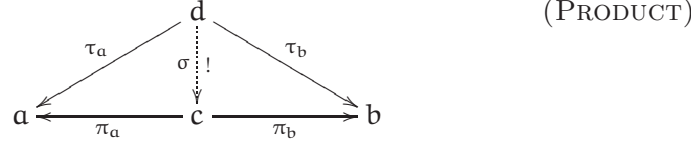
Definition 1.33 *Given objects a and b in \mathbf{K} . An object c is called the product of a and b iff*

1. *there exist morphisms $\pi_a : c \rightarrow a$ and $\pi_b : c \rightarrow b$,*
2. *for each object d and morphisms $\tau_a : d \rightarrow a$ and $\tau_b : d \rightarrow b$ there exists a unique morphism $\sigma : d \rightarrow c$ such that $\tau_a = \pi_a \circ \sigma$ and $\tau_b = \pi_b \circ \sigma$.*

Morphisms π_a and π_b are called projections to a resp. b .

Thus τ_a and τ_b factor uniquely through π_a and π_b . Note that we insist on having a unique factor, and that the factor should be the same for both pretenders. We will see in a minute why this is a sensible assumption. If it exists, the product of objects a and b is denoted by $a \times b$; the projections π_a and π_b are usually understood and not mentioned explicitly.

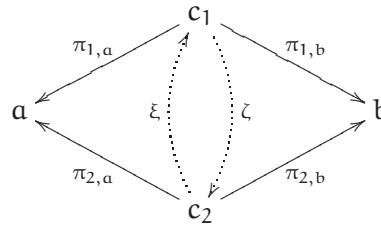
This diagram depicts the situation:



Lemma 1.34 *If the product of two objects exists, it is unique up to isomorphism.*

Proof Let a and b be the objects in question, also assume that c_1 and c_2 are products with morphisms $\pi_{i,a} \rightarrow a$ and $\pi_{i,b} \rightarrow b$ as the corresponding morphisms, $i = 1, 2$.

Because c_1 together with $\pi_{1,a}$ and $\pi_{1,b}$ is a product, we find a unique morphism $\xi : c_2 \rightarrow c_1$ with $\pi_{2,a} = \pi_{1,a} \circ \xi$ and $\pi_{2,b} = \pi_{1,b} \circ \xi$; similarly, we find a unique morphism $\zeta : c_1 \rightarrow c_2$ with $\pi_{1,a} = \pi_{2,a} \circ \zeta$ and $\pi_{1,b} = \pi_{2,b} \circ \zeta$.



Now look at $\xi \circ \zeta$: We obtain

$$\begin{aligned}\pi_{1,a} \circ \xi \circ \zeta &= \pi_{2,a} \circ \zeta = \pi_{1,a} \\ \pi_{1,b} \circ \xi \circ \zeta &= \pi_{2,b} \circ \zeta = \pi_{1,b}\end{aligned}$$

Then uniqueness of the factorization implies that $\xi \circ \zeta = \text{id}_{c_1}$, similarly, $\zeta \circ \xi = \text{id}_{c_2}$. Thus ξ and ζ are isomorphisms. \dashv

Let us have a look at some examples, first and foremost sets.

Example 1.35 Consider the category **Set** with maps as morphisms. Given sets A and B , we claim that $A \times B$ together with the projections $\pi_A : \langle a, b \rangle \mapsto a$ and $\pi_B : \langle a, b \rangle \mapsto b$ constitute the product of A and B in **Set**. In fact, if $\tau_A : D \rightarrow A$ and $\tau_B : D \rightarrow B$ are maps for some set D , then $\sigma : d \mapsto \langle \tau_A(d), \tau_B(d) \rangle$ satisfies the equations $\tau_A = \pi_A \circ \sigma$, $\tau_B = \pi_B \circ \sigma$, and it is clear that this is the only way to factor, so σ is uniquely determined. \mathbb{N}

If sets carry an additional structure, this demands additional attention.

Example 1.36 Let (S, \mathcal{A}) and (T, \mathcal{B}) be measurable spaces, so we are now in the category **Meas** of measurable spaces with measurable maps as morphisms, see Example 1.11. For constructing a product one is tempted to take the product $S \times T$ is **Set** and to find a suitable σ -algebra \mathcal{C} on $S \times T$ such that the projections π_S and π_T become measurable. Thus \mathcal{C} would have to contain $\pi_S^{-1}[A] = A \times T$ and $\pi_T^{-1}[B] = S \times B$ for each $A \in \mathcal{A}$ and each $B \in \mathcal{B}$. Because a σ -algebra is closed under intersections, \mathcal{C} would have to contain all measurable rectangles $A \times B$ with sides in \mathcal{A} and \mathcal{B} . So let's try this:

$$\mathcal{C} := \sigma(\{A \times B \mid A \in \mathcal{A}, B \in \mathcal{B}\})$$

Then clearly $\pi_S : (S \times T, \mathcal{C}) \rightarrow (S, \mathcal{A})$ and $\pi_T : (S \times T, \mathcal{C}) \rightarrow (T, \mathcal{B})$ are morphisms in **Meas**. Now let (D, \mathcal{D}) be a measurable space with morphisms $\tau_S : D \rightarrow S$ and $\tau_T : D \rightarrow T$, and define σ as above through $\sigma(d) := \langle \tau_S(d), \tau_T(d) \rangle$. We claim that σ is a morphism in **Meas**. It has to be shown that $\sigma^{-1}[C] \in \mathcal{D}$ for all $C \in \mathcal{C}$. We have a look at all elements of \mathcal{C} for which this is true, and we define

$$\mathcal{G} := \{C \in \mathcal{C} \mid \sigma^{-1}[C] \in \mathcal{D}\}.$$

If we can show that $\mathcal{G} = \mathcal{C}$, we are done. It is evident that \mathcal{G} is a σ -algebra, because the inverse image of a map respects countable Boolean operations. Moreover, if $A \in \mathcal{A}$ and $B \in \mathcal{B}$, then $\sigma^{-1}[A \times B] = \tau_S^{-1}[A] \cap \tau_T^{-1}[B] \in \mathcal{D}$, so that $A \times B \in \mathcal{G}$, provided $A \in \mathcal{A}, B \in \mathcal{B}$. But now we have

$$\mathcal{C} = \sigma(\{A \times B \mid A \in \mathcal{A}, B \in \mathcal{B}\}) \subseteq \mathcal{G} \subseteq \mathcal{C}.$$

Hence each element of \mathcal{C} is a member of \mathcal{G} , thus σ is \mathcal{D} - \mathcal{C} -measurable. Again, the construction shows that there is no other possibility for defining σ . Hence we have shown that two objects in the category **Meas** of measurable spaces with measurable maps have a product.

The σ -algebra \mathcal{C} which is constructed above is usually denoted by $\mathcal{A} \otimes \mathcal{B}$ and called the *product σ -algebra* of \mathcal{A} and \mathcal{B} . \mathbb{N}

Example 1.37 While the category **Meas** has products, the situation changes when taking probability measures into account, hence when changing to the category **Prob** of probability spaces, see Example 1.13. Recall that the product measure $\mu \otimes \nu$ of two probability measures μ on σ -algebra \mathcal{A} resp. ν on \mathcal{B} is the unique probability measure on the product σ -algebra $\mathcal{A} \otimes \mathcal{B}$ with $(\mu \otimes \nu)(A \times B) = \mu(A) \cdot \nu(B)$ for $A \in \mathcal{A}$ and $B \in \mathcal{B}$, in particular, $\pi_S : (S \times T, \mathcal{A} \otimes \mathcal{B}, \mu \otimes \nu) \rightarrow (S, \mathcal{A}, \mu)$ and $\pi_T : (S \times T, \mathcal{A} \otimes \mathcal{B}, \mu \otimes \nu) \rightarrow (T, \mathcal{B}, \nu)$ are morphisms in **Prob**.

Now define $S := T := [0, 1]$ and take in each case the smallest σ -algebra which is generated by the open intervals as a σ -algebra, hence put $\mathcal{A} := \mathcal{B} := \mathcal{B}([0, 1])$; λ is Lebesgue measure on


$\mathcal{B}([0, 1])$. Define

$$\kappa(E) := \lambda(\{x \in [0, 1] \mid \langle x, x \rangle \in E\})$$

for $E \in \mathcal{A} \otimes \mathcal{B}$ (well, we have to show that $\{x \in [0, 1] \mid \langle x, x \rangle \in E\} \in \mathcal{B}([0, 1])$, whenever $E \in \mathcal{A} \otimes \mathcal{B}$. This is not difficult and relegated to Exercise 10). Then $\pi_S : (S \times T, \mathcal{A} \otimes \mathcal{B}, \kappa) \rightarrow (S, \mathcal{A}, \lambda)$ and $\pi_T : (S \times T, \mathcal{A} \otimes \mathcal{B}, \kappa) \rightarrow (T, \mathcal{B}, \lambda)$ are morphisms in **Prob**, because $\kappa(\pi_S^{-1}[G]) = \kappa(G \times T) = \lambda(\{x \in [0, 1] \mid \langle x, x \rangle \in G \times T\}) = \lambda(G)$ for $G \in \mathcal{B}(S)$. If we could find a morphism $f : (S \times T, \mathcal{A} \otimes \mathcal{B}, \kappa) \rightarrow (S \times T, \mathcal{A} \otimes \mathcal{B}, \lambda \otimes \lambda)$ factoring through the projections, f would have to be the identity; thus would imply that $\kappa = \lambda \otimes \lambda$, but this is not the case: take $E := [1/2, 1] \times [0, 1/3]$, then $\kappa(E) = 0$, but $(\lambda \otimes \lambda)(E) = 1/6$.


Thus we conclude that the category **Prob** of probability spaces does not have products. 

The product topology on the Cartesian product of the carrier sets of topological spaces is familiar, open sets in the product just contain open rectangles. The categorical view is that of a product in the category of topological spaces.

Example 1.38 Let (T, \mathcal{G}) and (T, \mathcal{H}) be topological spaces, and equip the Cartesian product $S \times T$ with the product topology $\mathcal{G} \times \mathcal{H}$. This is the smallest topology on $S \times T$ which contains all the open rectangles $G \times H$ with $G \in \mathcal{G}$ and $H \in \mathcal{H}$. We claim that this is a product in the category **Top** of topological spaces. In fact, the projections $\pi_S : S \times T \rightarrow S$ and $\pi_T : S \times T \rightarrow T$ are continuous, because, e.g., $\pi_S^{-1}[G] = G \times T \in \mathcal{G} \times \mathcal{H}$. Now let (D, \mathcal{D}) be a topological space with continuous maps $\tau_S : D \rightarrow S$ and $\tau_T : D \rightarrow T$, and define $\sigma : D \rightarrow S \times T$ through $\sigma : d \mapsto \langle \tau_S(d), \tau_T(d) \rangle$. Then $\sigma^{-1}[G \times H] = \tau_S^{-1}[G] \cap \tau_T^{-1}[H] \in \mathcal{D}$, and since the inverse image of a topology under a map is a topology again, $\sigma : (D, \mathcal{D}) \rightarrow (S \times T, \mathcal{G} \times \mathcal{H})$ is continuous. Again, this is the only way to define a morphism σ so that $\tau_S = \pi_S \circ \sigma$ and $\tau_T = \pi_T \circ \sigma$. 

The category coming from a partially ordered set from Example 1.4 is investigated next.

Example 1.39 Let (P, \leq) be a partially ordered set, considered as a category **P**. Let $a, b \in P$, and assume that a and b have a product x in **P**. Thus there exist morphisms $\pi_a : x \rightarrow a$ and $\pi_b : x \rightarrow b$, which means by the definition of this category that $x \leq a$ and $x \leq b$ hold, hence that x is a lower bound to $\{a, b\}$. Moreover, if y is such that there are morphisms $\tau_a : y \rightarrow a$ and $\tau_b : y \rightarrow b$, then there exists a unique $\sigma : y \rightarrow x$ with $\tau_a = \pi_a \circ \sigma$ and $\tau_b = \pi_b \circ \sigma$. Translated into (P, \leq) , this means that if $y \leq a$ and $y \leq b$, then $y \leq x$ (morphisms in **Meas** are unique, if they exist). Hence the product x is just the greatest lower bound of $\{a, b\}$.

So the product corresponds to the infimum. This example demonstrates again that products do not necessarily exist in a category. 

Given morphisms $f : x \rightarrow a$ and $g : x \rightarrow b$, and assuming that the product $a \times b$ exists, we want to “lift” f and g to the product, i. e., we want to find a morphism $h : x \rightarrow a \times b$ with $f = \pi_a \circ h$ and $g = \pi_b \circ h$. Let us see how this is done in **Set**: Here $f : X \rightarrow A$ and $g : X \rightarrow B$ are maps, and one defines the lifted map $h : X \rightarrow A \times B$ through $h : x \mapsto \langle f(x), g(x) \rangle$, so that the conditions on the projections is satisfied. The next lemma states that this is always possible in a unique way.

Lemma 1.40 Assume that the product $a \times b$ exists for the objects a and b . Let $f : x \rightarrow a$ and $g : x \rightarrow b$ be morphisms. Then there exists a unique morphism $q : x \rightarrow a \times b$ such that $f = \pi_a \circ q$ and $g = \pi_b \circ q$. Morphism q is denoted by $f \times g$.

Proof The diagram looks like this:

$$\begin{array}{ccccc}
 & & x & & \\
 & f \swarrow & \downarrow q & \searrow g & \\
 a & \xleftarrow{\pi_a} & a \times b & \xrightarrow{\pi_b} & b
 \end{array}$$

Because $f : x \rightarrow a$ and $g : x \rightarrow b$, there exists a unique $q : x \rightarrow a \times b$ with $f = \pi_a \circ q$ and $g = \pi_b \circ q$. This follows from the definition of the product. \dashv

Let us look at the product through our $\text{hom}_{\mathbf{K}}$ -glasses. If $a \times b$ exists, and if $\tau_a : d \rightarrow a$ and $\tau_b : d \rightarrow b$ are morphisms, we know that there is a *unique* $\sigma : d \rightarrow a \times b$ rendering this diagram commutative

$$\begin{array}{ccccc}
 & & d & & \\
 & \tau_a \swarrow & \downarrow \sigma & \searrow \tau_b & \\
 a & \xleftarrow{\pi_a} & a \times b & \xrightarrow{\pi_b} & b
 \end{array}$$

Thus the map

$$p_d : \begin{cases} \text{hom}_{\mathbf{K}}(d, a) \times \text{hom}_{\mathbf{K}}(d, b) & \rightarrow \text{hom}_{\mathbf{K}}(d, a \times b) \\ \langle \tau_a, \tau_b \rangle & \mapsto \sigma \end{cases}$$

is well defined. In fact, we can say more

Proposition 1.41 p_d is a bijection.

Proof Assume $\sigma = p_d(f, g) = p_d(f', g')$. Then $f = \pi_a \circ \sigma = f'$ and $g = \pi_b \circ \sigma = g'$. Thus $\langle f, g \rangle = \langle f', g' \rangle$. Hence p_d is injective. Similarly, one shows that p_d is surjective: Let $h \in \text{hom}_{\mathbf{K}}(d, a \times b)$, then $\pi_a \circ h : d \rightarrow a$ and $\pi_b \circ h : d \rightarrow b$ are morphisms, so there exists a unique $h' : d \rightarrow a \times b$ with $\pi_a \circ h' = \pi_a \circ h$ and $\pi_b \circ h' = \pi_b \circ h$. Uniqueness implies that $h = h'$, so h occurs in the image of p_d . \dashv

Let us consider the dual construction.

Definition 1.42 Given objects a and b in category \mathbf{K} , the object s together with morphisms $i_a : a \rightarrow s$ and $i_b : b \rightarrow s$ is called the *coproduct* (or the *sum*) of a and b iff for each object t with morphisms $j_a : a \rightarrow t$ and $j_b : b \rightarrow t$ there exists a unique morphism $r : s \rightarrow t$ such that $j_a = r \circ i_a$ and $j_b = r \circ i_b$. Morphisms i_a and i_b are called *injections*, the coproduct of a and b is denoted by $a + b$.

This is the corresponding diagram:

$$\begin{array}{ccccc}
 & & t & & \\
 & j_a \swarrow & \uparrow r & \nwarrow j_b & \\
 a & \xrightarrow{i_a} & s & \xleftarrow{i_b} & b
 \end{array} \quad (\text{COPRODUCT})$$

Let us have a look at some examples.

Example 1.43 Let (P, \leq) be a partially ordered set, and consider category \mathbf{P} , as in Example 1.39. The coproduct of the elements a and b is just the supremum $\sup\{a, b\}$. This is

shown with exactly the same arguments which have been used in Example 1.39 for showing the the product of two elements corresponds to their infimum. \mathbb{M}

And then there is of course category **Set**.

Example 1.44 Let A and B be disjoint sets. Then $S := A \cup B$ together with

$$i_A : \begin{cases} A & \rightarrow S \\ a & \mapsto a \end{cases} \qquad i_B : \begin{cases} B & \rightarrow S \\ b & \mapsto b \end{cases}$$

form the coproduct of A and B . In fact, if T is a set with maps $j_A : A \rightarrow T$ and $j_B : B \rightarrow T$, then define

$$r : \begin{cases} S & \rightarrow T \\ s & \mapsto j_A(a), \text{ if } s = i_A(a), \\ s & \mapsto j_B(b), \text{ if } s = i_B(b) \end{cases}$$

Then $j_A = r \circ i_A$ and $j_B = r \circ i_B$, and these definitions are the only possible ones.

Note that we needed for this construction to work disjointness of the participating sets. Consider for example $A := \{-1, 0\}$, $B := \{0, 1\}$ and let $T := \{-1, 0, 1\}$ with $j_A(x) := -1$, $j_B(x) := +1$. No matter where we embed A and B , we cannot factor j_A and j_B uniquely.

If the sets are not disjoint, we first do a preprocessing step and embed them, so that the embedded sets are disjoint. The injections have to be adjusted accordingly. So this construction would work: Given sets A and B , define $S := \{\langle a, 1 \rangle \mid a \in A\} \cup \{\langle b, 2 \rangle \mid b \in B\}$ with $i_A : a \mapsto \langle a, 1 \rangle$ and $i_B : b \mapsto \langle b, 2 \rangle$. Note that we do not take a product like $S \times \{1\}$, but rather use a very specific construction; this is so since the product is determined uniquely only by isomorphism, so we might not have gained anything by using it. Of course, one has to be sure that the sum is not dependent in an essential way on this embedding. \mathbb{M}

The question of uniqueness is answered through this observation. It relates the coproduct in \mathbf{K} to the product in the dual category \mathbf{K}^{op} (see Example 1.17).

Proposition 1.45 *The coproduct s of objects a and b with injections $i_a : a \rightarrow s$ and $i_b : b \rightarrow s$ in category \mathbf{K} is the product in category \mathbf{K}^{op} with projections $i_a : s \rightarrow^{\text{op}} a$ and $i_b : s \rightarrow^{\text{op}} b$.*

Proof Revert in diagram (COPRODUCT) on page 19 to obtain diagram (PRODUCT) on page 16. \dashv

Corollary 1.46 *If the coproduct of two objects in a category exists, it is unique up to isomorphisms.*

Proof Proposition 1.45 together with Lemma 1.34. \dashv

Let us have a look at the coproduct for topological spaces.

Example 1.47 Given topological spaces (S, \mathcal{G}) and (T, \mathcal{H}) , we may and do assume that S and T are disjoint. Otherwise wrap the elements of the sets accordingly; put

$$\begin{aligned} A^\dagger &:= \{\langle a, 1 \rangle \mid a \in A\}, \\ B^\dagger &:= \{\langle b, 2 \rangle \mid b \in B\}, \end{aligned}$$

and consider the topological spaces $(S^\dagger, \{G^\dagger \mid G \in \mathcal{G}\})$ and $(T^\dagger, \{H^\dagger \mid H \in \mathcal{H}\})$ instead of (S, \mathcal{G}) and (T, \mathcal{H}) . Define on the coproduct $S + T$ of S and T in **Set** with injections i_S and i_T the topology

$$\mathcal{G} + \mathcal{H} := \{W \subseteq S + T \mid i_S^{-1}[W] \in \mathcal{G} \text{ and } i_T^{-1}[W] \in \mathcal{H}\}.$$

This is a topology: Both \emptyset and $S + T$ are members of $\mathcal{G} + \mathcal{H}$, and since \mathcal{G} and \mathcal{H} are topologies, $\mathcal{G} + \mathcal{H}$ is closed under finite intersections and arbitrary unions. Moreover, both $i_S : (S, \mathcal{G}) \rightarrow (S + T, \mathcal{G} + \mathcal{H})$ and $i_T : (T, \mathcal{H}) \rightarrow (S + T, \mathcal{G} + \mathcal{H})$ are continuous; in fact, $\mathcal{G} + \mathcal{H}$ is the smallest topology on $S + T$ with this property.

Now assume that $j_S : (S, \mathcal{G}) \rightarrow (R, \mathcal{R})$ and $j_T : (T, \mathcal{H}) \rightarrow (R, \mathcal{R})$ are continuous maps, and let $r : S + T \rightarrow R$ be the unique map determined by the coproduct in **Set**. Wouldn't it be nice if r is continuous? Actually, it is. Let $W \in \mathcal{R}$ be open in R , then $i_S^{-1}[r^{-1}[W]] = (r \circ i_S)^{-1}[W] = j_S^{-1}[W] \in \mathcal{G}$, similarly, $i_T^{-1}[r^{-1}[W]] \in \mathcal{H}$, thus by definition, $r^{-1}[W] \in \mathcal{G} + \mathcal{H}$. Hence we have found the factorization $j_S = r \circ i_S$ and $j_T = r \circ i_T$ in the category **Top**. This factorization is unique, because it is inherited from the unique factorization in **Set**. Hence we have shown that **Top** has finite coproducts. 👉

A similar construction applies to the category of measurable spaces.

Example 1.48 Let (S, \mathcal{A}) and (T, \mathcal{B}) be measurable spaces; we may assume again that the carrier sets S and T are disjoint. Take the injections $i_S : S \rightarrow S + T$ and $i_T : T \rightarrow S + T$ from **Set**. Then

$$\mathcal{A} + \mathcal{B} := \{W \subseteq S + T \mid i_S^{-1}[W] \in \mathcal{A} \text{ and } i_T^{-1}[W] \in \mathcal{B}\}$$

is a σ -algebra, $i_S : (S, \mathcal{A}) \rightarrow (S + T, \mathcal{A} + \mathcal{B})$ and $i_T : (T, \mathcal{B}) \rightarrow (S + T, \mathcal{A} + \mathcal{B})$ are measurable. The unique factorization property is established in exactly the same way as for **Top**. 👉

Example 1.49 Let us consider the category **Rel** of relations, which is based on sets as objects. If S and T are sets, we again may and do assume that they are disjoint. Then $S + T = S \cup T$ together with the injections

$$\begin{aligned} I_S &:= \{\langle s, i_S(s) \rangle \mid s \in S\}, \\ I_T &:= \{\langle t, i_T(t) \rangle \mid t \in T\} \end{aligned}$$

form the coproduct, where i_S and i_T are the injections into $S + T$ from **Set**. In fact, we have to show that we can find for given relations $q_S \subseteq S \times D$ and $q_T \subseteq T \times D$ a unique relation $Q \subseteq (S + T) \times D$ with $q_S = I_S \circ Q$ and $q_T = I_T \circ Q$. The choice is fairly straightforward: Define

$$Q := \{\langle i_S(s), q \rangle \mid \langle s, q \rangle \in q_S\} \cup \{\langle i_T(t), q \rangle \mid \langle t, q \rangle \in q_T\}.$$

Thus

$$\langle s, q \rangle \in I_S \circ Q \Leftrightarrow \text{there exists } x \text{ with } \langle s, x \rangle \in I_S \text{ and } \langle x, q \rangle \in Q \Leftrightarrow \langle s, q \rangle \in q_S.$$

Hence $q_S = I_S \circ Q$, similarly, $q_T = I_T \circ Q$. It is clear that no other choice is possible.

Consequently, the coproduct is the same as in **Set**. 👉

We have just seen in a simple example that dualizing, i.e., going to the dual category, is very helpful. Instead of proving directly that the coproduct is uniquely determined up to isomorphism, if it exists, we turned to the dual category and reused the already established

result that the product is uniquely determined, casting it into a new context. The duality, however, is a purely structural property, it usually does not help us with specific constructions. This could be seen when we wanted to construct the coproduct of two sets; it did not help here that we knew how to construct the product of two sets, even though product and coproduct are intimately related through dualization. We will make the same observation when we deal with pullbacks and pushouts.

1.2.2 Pullbacks and Pushouts

Sometimes one wants to complete the square as in the diagram below on the left hand side:



Hence one wants to find an object d together with morphisms $i_1 : d \rightarrow a$ and $i_2 : d \rightarrow b$ rendering the diagram on the right hand side commutative. This completion should be as coarse as possible in this sense. If we have another objects, say, e with morphisms $j_1 : e \rightarrow a$ and $j_2 : e \rightarrow b$ such that $f \circ j_1 = g \circ j_2$, then we want to be able to uniquely factor through i_1 and i_2 .

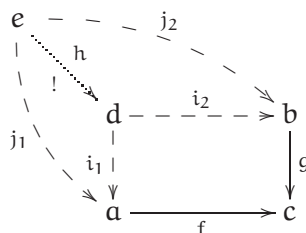
This is captured in the following definition.

Definition 1.50 Let $f : a \rightarrow c$ and $g : b \rightarrow c$ be morphisms in \mathbf{K} with the same codomain. An object d together with morphisms $i_1 : d \rightarrow a$ and $i_2 : d \rightarrow b$ is called a pullback of f and g iff

1. $f \circ i_1 = g \circ i_2$,
2. If e is an object with morphisms $j_1 : e \rightarrow a$ and $j_2 : e \rightarrow b$ such that $f \circ j_1 = g \circ j_2$, then there exists a unique morphism $h : e \rightarrow d$ such that $j_1 = i_1 \circ h$ and $j_2 = i_2 \circ h$.

If we postulate the existence of the morphism $h : e \rightarrow d$, but do not insist on its uniqueness, then d with i_1 and i_2 is called a weak pullback.

A diagram for a pullback looks like this



It is clear that a pullback is unique up to isomorphism; this is shown in exactly the same way as in Lemma 1.34. Let us have a look at **Set** as an important example to get a first impression on the inner workings of a pullback.

Example 1.51 Let $f : X \rightarrow Z$ and $g : Y \rightarrow Z$ be maps. We claim that


$$P := \{\langle x, y \rangle \in X \times Y \mid f(x) = g(y)\}$$

together with the projections $\pi_X : \langle x, y \rangle \mapsto x$ and $\pi_Y : \langle x, y \rangle \mapsto y$ is a pullback for f and g .

Let $\langle x, y \rangle \in P$, then

$$(f \circ \pi_X)(x, y) = f(x) = g(y) = (g \circ \pi_Y)(x, y),$$

so that the first condition is satisfied. Now assume that $j_X : T \rightarrow X$ and $j_Y : T \rightarrow Y$ satisfies $f(j_X(t)) = g(j_Y(t))$ for all $t \in T$. Thus $\langle j_X(t), j_Y(t) \rangle \in P$ for all t , and defining $r(t) := \langle j_X(t), j_Y(t) \rangle$, we obtain $j_X = \pi_X \circ r$ and $j_Y = \pi_Y \circ r$. Moreover, this is the only possibility to define a factor map with the desired property.

An interesting special case occurs for $X = Y$ and $f = g$. Then $P = \ker(f)$, so that the kernel of a map occurs as a pullback in category **Set**. 

As an illustration for the use of a pullback construction, look at this simple statement.

Lemma 1.52 Assume that d with morphisms $i_a : d \rightarrow a$ and $i_b : d \rightarrow b$ is a pullback for $f : a \rightarrow c$ and $g : b \rightarrow c$. If g is a mono, so is i_a .

Proof Let $g_1, g_2 : e \rightarrow d$ be morphisms with $i_a \circ g_1 = i_a \circ g_2$. We have to show that $g_1 = g_2$ holds. If we know that $i_b \circ g_1 = i_b \circ g_2$, we may use the definition of a pullback and capitalize on the uniqueness of the factorization. But let's see.

From $i_a \circ g_1 = i_b \circ g_2$ we conclude $f \circ i_a \circ g_1 = f \circ i_b \circ g_2$, and because $f \circ i_a = g \circ i_b$, we obtain $g \circ i_b \circ g_1 = g \circ i_b \circ g_2$. Since g is a mono, we may cancel on the left of this equation, and we obtain, as desired, $i_b \circ g_1 = i_b \circ g_2$.

But since we have a pullback, there exists a unique $h : e \rightarrow d$ with $i_a \circ g_1 = i_a \circ h$ ($= i_a \circ g_2$) and $i_b \circ g_1 = i_b \circ h$ ($= i_b \circ g_2$). We see that the morphisms g_1, g_2 and h have the same properties with respect to factoring, so they must be identical by uniqueness. Hence $g_1 = h = g_2$, and we are done. \dashv

This is another simple example for the use of a pullback in **Set**.

Example 1.53 Let R be an equivalence relation on a set X with projections $\pi_1 : \langle x_1, x_2 \rangle \mapsto x_1$; the second projection $\pi_2 : R \rightarrow X$ is defined similarly. Then

$$\begin{array}{ccc} R & \xrightarrow{\pi_2} & X \\ \pi_1 \downarrow & & \downarrow \eta_R \\ X & \xrightarrow{\eta_R} & X/R \end{array}$$

(with $\eta_R : x \mapsto [x]_R$) is a pullback diagram. In fact, the diagram commutes. Let $\alpha, \beta : M \rightarrow X$ be maps with $\alpha \circ \eta_R = \beta \circ \eta_R$, thus $[\alpha(m)]_R = [\beta(m)]_R$ for all $m \in M$; hence $\langle \alpha(m), \beta(m) \rangle \in R$ for all m . The only map $\tau : M \rightarrow R$ with $\alpha = \pi_1 \circ \tau$ and $\beta = \pi_2 \circ \tau$ is $\tau(m) := \langle \alpha(m), \beta(m) \rangle$.



Pullbacks are compatible with products in a sense which we will make precise in a moment. Before we do that, however, we need an auxiliary statement:

Lemma 1.54 *Assume that the products $\mathbf{a} \times \mathbf{a}'$ and $\mathbf{b} \times \mathbf{b}'$ exist in category \mathbf{K} . Given morphisms $f : \mathbf{a} \rightarrow \mathbf{b}$ and $f' : \mathbf{a}' \rightarrow \mathbf{b}'$, there exists a unique morphism $f \times f' : \mathbf{a} \times \mathbf{a}' \rightarrow \mathbf{b} \times \mathbf{b}'$ such that*

$$\begin{aligned}\pi_{\mathbf{b}} \circ f \times f' &= f \circ \pi_{\mathbf{a}} \\ \pi_{\mathbf{b}'} \circ f \times f' &= f' \circ \pi_{\mathbf{a}'}\end{aligned}$$

Proof Apply the definition of a product to the morphisms $f \circ \pi_{\mathbf{a}} : \mathbf{a} \times \mathbf{a}' \rightarrow \mathbf{b}$ and $f' \circ \pi_{\mathbf{a}'} : \mathbf{a} \times \mathbf{a}' \rightarrow \mathbf{b}'$. \dashv

Thus the morphism $f \times f'$ constructed in the lemma renders both parts of this diagram commutative.

$$\begin{array}{ccccc}\mathbf{a} & \xleftarrow{\pi_{\mathbf{a}}} & \mathbf{a} \times \mathbf{a}' & \xrightarrow{\pi_{\mathbf{a}'}} & \mathbf{a}' \\ f \downarrow & & f \times f' \downarrow & & \downarrow f' \\ \mathbf{b} & \xleftarrow{\pi_{\mathbf{b}}} & \mathbf{b} \times \mathbf{b}' & \xrightarrow{\pi_{\mathbf{b}'}} & \mathbf{b}'\end{array}$$

Denoting this morphism as $f \times f'$, we note that \times is overloaded for morphisms, a look at domains and codomains indicates without ambiguity, however, which version is intended.

Quite apart from its general interest, this is what we need Lemma 1.54 for.

Lemma 1.55 *Assume that we have these pullbacks*

$$\begin{array}{ccc} \mathbf{a} & \xrightarrow{f} & \mathbf{b} \\ g \downarrow & & \downarrow h \\ \mathbf{c} & \xrightarrow{k} & \mathbf{d} \end{array} \qquad \begin{array}{ccc} \mathbf{a}' & \xrightarrow{f'} & \mathbf{b}' \\ g' \downarrow & & \downarrow h' \\ \mathbf{c}' & \xrightarrow{k'} & \mathbf{d}' \end{array}$$

Then this is a pullback diagram as well

$$\begin{array}{ccc} \mathbf{a} \times \mathbf{a}' & \xrightarrow{f \times f'} & \mathbf{b} \times \mathbf{b}' \\ g \times g' \downarrow & & \downarrow h \times h' \\ \mathbf{c} \times \mathbf{c}' & \xrightarrow{k \times k'} & \mathbf{d} \times \mathbf{d}' \end{array}$$

Proof 1. We show first that the diagram commutes. It is sufficient to compute the projections, from uniqueness then equality will follow. Allora:

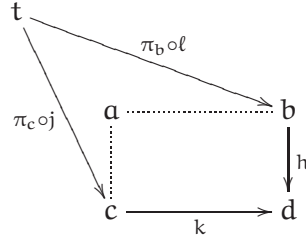
$$\begin{aligned}\pi_{\mathbf{d}} \circ (h \times h') \circ (f \times f') &= (h \circ \pi_{\mathbf{a}}) \circ (f \times f') = h \circ f \circ \pi_{\mathbf{a}} \\ \pi_{\mathbf{d}} \circ (k \times k') \circ (g \times g') &= k \circ \pi_{\mathbf{c}} \circ (g \times g') = k \circ g \circ \pi_{\mathbf{c}} = h \circ f \circ \pi_{\mathbf{a}}.\end{aligned}$$

A similar computation is carried out for $\pi_{\mathbf{d}'}$.

2. Let $j : \mathbf{t} \rightarrow \mathbf{c} \times \mathbf{c}'$ and $\ell : \mathbf{t} \rightarrow \mathbf{b} \times \mathbf{b}'$ be morphisms such that $(k \times k') \circ j = (h \times h') \circ \ell$, then we claim that there exists a unique morphism $r : \mathbf{t} \rightarrow \mathbf{a} \times \mathbf{a}'$ such that $j = (g \times g') \circ r$

and $\ell = (f \times f') \circ r$. The plan is to obtain r from the projections and then show that this morphism is unique.

3. We show that this diagram commutes



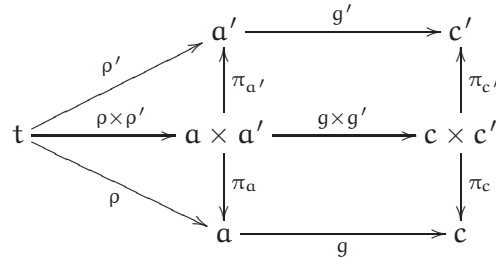
We have

$$\begin{aligned}
 k \circ (\pi_c \circ j) &= (k \circ \pi_c) \circ j \\
 &= (\pi_d \circ k \times k') \circ j \\
 &= \pi_d \circ (k \times k' \circ j) \\
 &\stackrel{(\ddagger)}{=} \pi_d \circ (h \times h' \circ \ell) \\
 &= (\pi_d \circ h \times h') \circ \ell \\
 &= (h \circ \pi_b) \circ \ell \\
 &= h \circ (\pi_b \circ \ell)
 \end{aligned}$$

In (\ddagger) we use Lemma 1.54. Using the primed part of Lemma 1.54 we obtain $k' \circ (\pi_{c'} \circ j) = h' \circ (\pi_{b'} \circ \ell)$.

Because the left hand side in the assumption is a pullback diagram, there exists a unique morphism $\rho : t \rightarrow a$ with $\pi_c \circ j = g \circ \rho$, $\pi_b \circ \ell = f \circ \rho$. Similarly, there exists a unique morphism $\rho' : t \rightarrow a'$ with $\pi_{c'} \circ j = g' \circ \rho'$, $\pi_{b'} \circ \ell = f' \circ \rho'$.

4. Put $r := \rho \times \rho'$, then $r : t \rightarrow a \times a'$, and we have this diagram



Hence

$$\begin{aligned}
 \pi_c \circ (g \times g') \circ (\rho \times \rho') &= g \circ \pi_a \circ (\rho \times \rho') = g \circ \rho = \pi_c \circ j \\
 \pi_{c'} \circ (g \times g') \circ (\rho \times \rho') &= g' \circ \pi_{a'} \circ (\rho \times \rho') = g' \circ \rho' = \pi_{c'} \circ j.
 \end{aligned}$$

Because a morphism into a product is uniquely determined by its projections, we conclude that $(g \times g') \circ (\rho \times \rho') = j$. Similarly, we obtain $(f \times f') \circ (\rho \times \rho') = \ell$.

5. Thus $r = \rho \times \rho'$ can be used for factoring; it remains to show that this is the only possible choice. In fact, let $\sigma : t \rightarrow a \times a'$ be a morphism with $(g \times g') \circ \sigma = j$ and $(f \times f') \circ \sigma = \ell$, then it is enough to show that $\pi_a \circ \sigma$ has the same properties as ρ , and that $\pi_{a'} \circ \sigma$ has the same properties as ρ' . Calculating the composition with g resp. f , we obtain

$$\begin{aligned} g \circ \pi_a \circ \sigma &= \pi_c \circ (g \times g') \circ \sigma = \pi_c \circ j \\ f \circ \pi_a \circ \sigma &= \pi_d \circ (f \times f') \circ \sigma = \pi_b \circ \ell \end{aligned}$$

This implies $\pi_a \circ \sigma = \rho$ by uniqueness of ρ , the same argument implies $\pi_{a'} \circ \sigma = \rho'$. But this means $\sigma = \rho \times \rho'$, and uniqueness is established. \dashv

Let's dualize. The pullback was defined so that the upper left corner of a diagram is filled in an essentially unique way, the dual construction will have to fill the lower right corner of a diagram in the same way. But by reversing arrows, we convert a diagram in which the lower right corner is missing into a diagram without an upper left corner:

$$\begin{array}{ccc} a & \longrightarrow & c \\ \downarrow & & \\ b & & \end{array} \quad \Rightarrow \quad \begin{array}{ccc} & & c \\ & & \downarrow \\ b & \longrightarrow & a \end{array}$$

The corresponding construction is called a pushout.

Definition 1.56 Let $f : a \rightarrow b$ and $g : a \rightarrow c$ be morphisms in category \mathbf{K} with the same domain. An object d together with morphisms $p_b : b \rightarrow d$ and $p_c : c \rightarrow d$ is called the pushout of f and g iff these conditions are satisfied:

1. $p_b \circ f = p_c \circ g$
2. if $q_b : b \rightarrow e$ and $q_c : c \rightarrow e$ are morphisms such that $q_b \circ f = q_c \circ g$, then there exists a unique morphism $h : d \rightarrow e$ such that $q_b = h \circ p_b$ and $q_c = h \circ p_c$.

This diagram obviously looks like this:

$$\begin{array}{ccc} a & \xrightarrow{g} & b \\ f \downarrow & & \downarrow p_b \\ c & \xrightarrow{p_c} & d \end{array} \quad \begin{array}{c} \text{---} q_b \text{---} \\ \text{---} q_c \text{---} \\ \text{---} h \text{---} \end{array} \quad \begin{array}{c} \text{---} e \\ \text{---} e \\ \text{---} e \end{array}$$

It is clear that the pushout of $f \in \text{hom}_{\mathbf{K}}(a, b)$ and $g \in \text{hom}_{\mathbf{K}}(a, c)$ is the pullback of $f \in \text{hom}_{\mathbf{K}^{\text{op}}}(\mathbf{b}, \mathbf{a})$ and of $g \in \text{hom}_{\mathbf{K}^{\text{op}}}(\mathbf{c}, \mathbf{a})$ in the dual category. This, however, does not really provide assistance when constructing a pushout. Let us consider specifically the category **Set** of sets with maps as morphisms. We know that dualizing a product yields a sum, but it is not quite clear how to proceed further. The next example tells us what to do.

Example 1.57 We are in the category **Set** of sets with maps as morphisms now. Consider maps $f : A \rightarrow B$ and $g : A \rightarrow C$. Construct on the sum $B + C$ the smallest equivalence relation R which contains $R_0 := \{ \langle (i_B \circ f)(a), (i_C \circ g)(a) \rangle \mid a \in A \}$. Here i_B and i_C are the injections

of B resp. C into the sum. Let D the factor space $(A + B)/R$ with $p_B : b \mapsto [i_B(b)]_R$ and $p_C : c \mapsto [i_C(c)]_R$. The construction yields $p_B \circ f = p_C \circ g$, because R identifies the embedded elements $f(a)$ and $g(a)$ for any $a \in A$.

Now assume that $q_B : B \rightarrow E$ and $q_C : C \rightarrow E$ are maps with $q_B \circ f = q_C \circ g$. Let $q : D \rightarrow E$ be the unique map with $q \circ i_B = q_B$ and $q \circ i_C = q_C$ (Lemma 1.40 together with Proposition 1.45). Then $R_0 \subseteq \ker(q)$: Let $a \in A$, then

$$q(i_B(f(a))) = q_B(f(a)) = q_C(g(a)) = q(i_C(f(a))),$$

so that $\langle i_B(f(a)), i_C(f(a)) \rangle \in \ker(q)$. Because $\ker(q)$ is an equivalence relation on D , we conclude $R \subseteq \ker(q)$. Thus $h([x]_R) := q(x)$ defines a map $D/R \rightarrow E$ with

$$h(p_B(b)) = h([i_B(b)]_R) = q(i_B(b)) = q_B(b),$$

$$h(p_C(c)) = h([i_C(c)]_R) = q(i_C(c)) = q_C(c)$$

for $b \in B$ and $c \in C$. It is clear that there is no other way to define a map h with the desired properties. ☞

So we have shown that the pushout in the category **Set** of sets with maps exists. To illustrate the construction, consider the pushout of two factor maps. In this example $\rho \vee \tau$ denotes the smallest equivalence relation which contains the equivalence relations ρ and τ .

Example 1.58 Let ρ and τ be equivalence relations on a set X with factor maps $\eta_\rho : X \rightarrow X/\rho$ and $\eta_\tau : X \rightarrow X/\tau$. Then the pushout of these maps is $X/(\rho \vee \tau)$ with $\zeta_\rho : [x]_\rho \mapsto [x]_{\rho \vee \tau}$ and $\zeta_\tau : [x]_\tau \mapsto [x]_{\rho \vee \tau}$ as the associated maps. In fact, we have $\eta_\rho \circ \zeta_\rho = \eta_\tau \circ \zeta_\tau$, so the first property is satisfied. Now let $t_\rho : X/\rho \rightarrow E$ and $t_\tau : X/\tau \rightarrow E$ be maps with $t_\rho \circ \eta_\rho = t_\tau \circ \eta_\tau$ for a set E , then $h : [x]_{\rho \vee \tau} \mapsto t_\rho([x]_\rho)$ maps $X/(\rho \vee \tau)$ to E with plainly $t_\rho = h \circ \zeta_\rho$ and $t_\tau = h \circ \zeta_\tau$; moreover, h is uniquely determined by this property. Because the pushout is up to isomorphism uniquely determined by Lemma 1.34 and Proposition 1.45, we have shown that the supremum of two equivalence relations in the lattice of equivalence relations can be computed through the pushout of its components. ☞

1.3 Functors and Natural Transformations

We introduce functors which help in transporting information between categories in a way similar to morphisms, which are thought to transport information between objects. Of course, we will have to observe some properties in order to capture in a formal way the intuitive understanding of a functor as a structure preserving element. Functors themselves can be related, leading to the notion of a natural transformation. Given a category, there is a plethora of functors and natural transformations provided by the hom sets; this is studied in some detail, first, because it is a built-in in every category, second because the Yoneda Lemma relates this rich structure to set based functors, which in turn will be used when studying adjunctions.

1.3.1 Functors

Loosely speaking, a functor is a pair of structure preserving maps between categories: it maps one category to another one in a compatible way. A bit more precise, a functor \mathbf{F} between

categories \mathbf{K} and \mathbf{L} assigns to each object \mathbf{a} in category \mathbf{K} an object $\mathbf{F}(\mathbf{a})$ in \mathbf{L} , and it assigns each morphism $f : \mathbf{a} \rightarrow \mathbf{b}$ in \mathbf{K} a morphism $\mathbf{F}(f) : \mathbf{F}(\mathbf{a}) \rightarrow \mathbf{F}(\mathbf{b})$ in \mathbf{L} ; some obvious properties have to be observed. In this way it is possible to compare categories, and to carry properties from one category to another one. To be more specific:

Definition 1.59 A functor $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$ assigns to each object \mathbf{a} in category \mathbf{K} an object $\mathbf{F}(\mathbf{a})$ in category \mathbf{L} and maps each hom set $\text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{b})$ of \mathbf{K} to the hom set $\text{hom}_{\mathbf{L}}(\mathbf{F}(\mathbf{a}), \mathbf{F}(\mathbf{b}))$ of \mathbf{L} subject to these conditions

- $\mathbf{F}(\text{id}_{\mathbf{a}}) = \text{id}_{\mathbf{F}(\mathbf{a})}$ for each object \mathbf{a} of \mathbf{K} ,
- if $f : \mathbf{a} \rightarrow \mathbf{b}$ and $g : \mathbf{b} \rightarrow \mathbf{c}$ are morphisms in \mathbf{K} , then $\mathbf{F}(g \circ f) = \mathbf{F}(g) \circ \mathbf{F}(f)$.

A functor $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{K}$ is called an *endofunctor* on \mathbf{K} .

The first condition says that the identity morphisms in \mathbf{K} are mapped to the identity morphisms in \mathbf{L} , and the second condition tell us that \mathbf{F} has to be compatible with composition in the respective categories. Note that for specifying a functor, we have to say what the functor does with objects, and how the functor transforms morphisms. By the way, we often write $\mathbf{F}(\mathbf{a})$ as $\mathbf{F}\mathbf{a}$, and $\mathbf{F}(f)$ as $\mathbf{F}f$.

Let us have a look at some examples. Trivial examples for functors include the *identity functor* $\text{Id}_{\mathbf{K}}$, which maps objects resp. morphisms to itself, and the *constant functor* $\Delta_{\mathbf{x}}$ for an object \mathbf{x} , which maps every object to \mathbf{x} , and every morphism to $\text{id}_{\mathbf{x}}$.

Example 1.60 Consider the category **Set** of sets with maps as morphisms. Given set X , $\mathcal{P}X$ is a set again; define $\mathcal{P}(f)(A) := f[A]$ for the map $f : X \rightarrow Y$ and for $A \subseteq X$, then $\mathcal{P}f : \mathcal{P}X \rightarrow \mathcal{P}Y$. We check the laws for a functor:

- $\mathcal{P}(\text{id}_X)(A) = \text{id}_X[A] = A = \text{id}_{\mathcal{P}X}(A)$, so that $\mathcal{P}\text{id}_X = \text{id}_{\mathcal{P}X}$.
- let $f : X \rightarrow Y$ and $g : Y \rightarrow Z$, then $\mathcal{P}f : \mathcal{P}X \rightarrow \mathcal{P}Y$ and $\mathcal{P}g : \mathcal{P}Y \rightarrow \mathcal{P}Z$ with

$$\begin{aligned} (\mathcal{P}(g) \circ \mathcal{P}(f))(A) &= \mathcal{P}(g)(\mathcal{P}(f)(A)) \\ &= g[f[A]] \\ &= \{g(f(\mathbf{a})) \mid \mathbf{a} \in A\} \\ &= (g \circ f)[A] \\ &= \mathcal{P}(g \circ f)(A) \end{aligned}$$

for $A \subseteq X$. Thus the *power set functor* \mathcal{P} is compatible with composition of maps.



Example 1.61 Given a category \mathbf{K} and an object \mathbf{a} of \mathbf{K} , associate

$$\begin{aligned} \mathbf{a}_+ : \mathbf{x} &\mapsto \text{hom}_{\mathbf{K}}(\mathbf{a}, \mathbf{x}) \\ \mathbf{a}^+ : \mathbf{x} &\mapsto \text{hom}_{\mathbf{K}}(\mathbf{x}, \mathbf{a}). \end{aligned}$$

with \mathbf{a} together with the maps on hom-sets $\text{hom}_{\mathbf{K}}(\mathbf{a}, \cdot)$ resp. $\text{hom}_{\mathbf{K}}(\cdot, \mathbf{a})$. Then \mathbf{a}^+ is a functor $\mathbf{K} \rightarrow \mathbf{Set}$.

In fact, given morphism $f : x \rightarrow y$, we have $a_+ f : \text{hom}_{\mathbf{K}}(a, x) \rightarrow \text{hom}_{\mathbf{K}}(a, y)$, taking g into $f \circ g$. Plainly, $a_+(\text{id}_x) = \text{id}_{\text{hom}_{\mathbf{K}}(a, x)} = \text{id}_{a_+(x)}$, and

$$a_+(g \circ f)(h) = (g \circ f) \circ h = g \circ (f \circ h) = a_+(g)(a_+(f)(h)),$$

if $f : x \rightarrow y, g : y \rightarrow z$ and $h : a \rightarrow x$. ✎

Functors come in handy when we want to forget part of the structure.

Example 1.62 Let **Meas** be the category of measurable spaces. Assign to each measurable space (X, \mathcal{C}) its carrier set X , and to each morphism $f : (X, \mathcal{C}) \rightarrow (Y, \mathcal{D})$ the corresponding map $f : X \rightarrow Y$. It is immediately checked that this constitutes a functor **Meas** \rightarrow **Set**. Similarly, we might forget the topological structure by assigning each topological space its carrier set, and assign each continuous map to itself. These functors are sometimes called *forgetful functors*. ✎

The following example twists Example 1.62 a little bit.

Example 1.63 Assign to each measurable space (X, \mathcal{C}) its σ -algebra $\mathbf{B}(X, \mathcal{C}) := \mathcal{C}$. Let $f : (X, \mathcal{C}) \rightarrow (Y, \mathcal{D})$ be a morphism in **Meas**; put $\mathbf{B}(f) := f^{-1}$, then $\mathbf{B}(f) : \mathbf{B}(Y, \mathcal{D}) \rightarrow \mathbf{B}(X, \mathcal{C})$, because f is \mathcal{C} - \mathcal{D} -measurable. We plainly have $\mathbf{B}(\text{id}_{X, \mathcal{C}}) = \text{id}_{\mathbf{B}(X, \mathcal{C})}$ and $\mathbf{B}(g \circ f) = (g \circ f)^{-1} = f^{-1} \circ g^{-1} = \mathbf{B}(f) \circ \mathbf{B}(g)$, so $\mathbf{B} : \mathbf{Meas} \rightarrow \mathbf{Set}$ is no functor, although it behaves like one. DON'T PANIC! If we reverse arrows, things work out properly: $\mathbf{B} : \mathbf{Meas} \rightarrow \mathbf{Set}^{\text{op}}$ is, as we have just shown, a functor (the dual \mathbf{K}^{op} of a category \mathbf{K} has been introduced in Example 1.17).

This functor could be called the *Borel functor* (the measurable sets are sometimes also called the Borel sets). ✎

Definition 1.64 A functor $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}^{\text{op}}$ is called a *contravariant functor between \mathbf{K} and \mathbf{L}* ; in contrast, a functor according to Definition 1.59 is called *covariant*.

If we talk about functors, we always mean the covariant flavor, contravariance is mentioned explicitly.

Let us complete the discussion from Example 1.61 by considering a^+ , which takes $f : x \rightarrow y$ to $a^+ f : \text{hom}_{\mathbf{K}}(y, a) \rightarrow \text{hom}_{\mathbf{K}}(x, a)$ through $g \mapsto g \circ f$. a^+ maps the identity on x to the identity on $\text{hom}_{\mathbf{K}}(x, a)$. If $g : y \rightarrow z$, we have

$$a^+(f)(a^+(g)(h)) = a^+(f)(h \circ g) = h \circ g \circ f = a^+(g \circ f)(h)$$

for $h : z \rightarrow a$. Thus a^+ is a contravariant functor $\mathbf{K} \rightarrow \mathbf{Set}$, while its cousin a_+ is covariant.

Functors may also be used to model structures.

Example 1.65 Consider this functor $\mathbf{S} : \mathbf{Set} \rightarrow \mathbf{Set}$ which assigns each set X the set $X^{\mathbb{N}}$ of all sequences over X ; the map $f : X \rightarrow Y$ is assigned the map $\mathbf{S} : (x_n)_{n \in \mathbb{N}} \mapsto (f(x_n))_{n \in \mathbb{N}}$. Evidently, id_X is mapped to $\text{id}_{X^{\mathbb{N}}}$, and it is easily checked that $\mathbf{S}(g \circ f) = \mathbf{S}(g) \circ \mathbf{S}(f)$. Hence \mathbf{S} constitutes an endofunctor on **Set**. ✎

Example 1.66 Similarly, define the endofunctor \mathbf{F} on **Set** by assigning X to $X^{\mathbb{N}} \cup X^*$ with X^* as the set of all finite sequences over X . Then $\mathbf{F}X$ has all finite or infinite sequences over the set X . Let $f : X \rightarrow Y$ be a map, and let $(x_i)_{i \in \mathbb{I}} \in \mathbf{F}X$ be a finite or infinite sequence, then put

$(\mathbf{F}f)(x_i)_{i \in I} := (f(x_i))_{i \in I} \in \mathbf{F}Y$. It is not difficult to see that \mathbf{F} satisfies the laws for a functor.

✎

The next example deals with automata which produce an output (in contrast to Example 1.15 where we mainly had state transitions in view).

Example 1.67 An *automaton with output* (A, B, X, δ) has an input alphabet A , an output alphabet B and a set X of states with a map $\delta : X \times A \rightarrow X \times B$; $\delta(x, a) = \langle x', b \rangle$ yields the next state x' and the output b if the input is a in state x . A morphism $f : (X, A, B, \delta) \rightarrow (Y, A, B, \tau)$ of automata is a map $f : X \rightarrow Y$ such that $\tau(f(x), a) = (f \times \text{id}_B)(\delta(x, a))$ for all $x \in X, a \in A$, thus $(f \times \text{id}_B) \circ \delta = \tau \circ (f \times \text{id}_A)$. This yields apparently a category \mathbf{AutO} , the category of automata with output.

We want to expose the state space X in order to make it a parameter to an automata, because input and output alphabets are given from the outside, so for modeling purposes only states are at our disposal. Hence we reformulate δ and take it as a map $\delta_* : X \rightarrow (X \times B)^A$ with $\delta_*(x)(a) := \delta(x, a)$. Now $f : (X, A, B, \delta) \rightarrow (Y, A, B, \tau)$ is a morphism iff this diagram commutes

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ \delta_* \downarrow & & \downarrow \tau_* \\ (X \times B)^A & \xrightarrow{f^\bullet} & (Y \times B)^A \end{array}$$

with $f^\bullet(t)(a) := (f \times \text{id}_B)(t(a))$. Let's see why this is the case. Given $x \in X, a \in A$, we have

$$f^\bullet(\delta_*(x))(a) = (f \times \text{id}_B)(\delta_*(x)(a)) = (f \times \text{id}_B)(\delta(x, a)) = \tau(f(x), a) = \tau_*(f(x))(a),$$

thus $f^\bullet(\delta_*(x)) = \tau_*(f(x))$ for all $x \in X$, hence $f^\bullet \circ \delta_* = \tau_* \circ f$, so the diagram is commutative indeed. Define $\mathbf{F}(X) := (X \times A)^B$, for an object (A, B, X, δ) in category \mathbf{AutO} and put $\mathbf{F}(f) := f^\bullet$ for the automaton morphism $f : (X, A, B, \delta) \rightarrow (Y, A, B, \tau)$, thus $\mathbf{F}(f)$ renders this diagram commutative:

$$\begin{array}{ccc} & A & \\ t \swarrow & & \searrow \mathbf{F}(f)(t) \\ X \times B & \xrightarrow{f \times \text{id}_B} & Y \times B \end{array}$$

We claim that $\mathbf{F} : \mathbf{AutO} \rightarrow \mathbf{Set}$ is functor. Let $g : (Y, A, B, \tau) \rightarrow (Z, A, B, \theta)$ be a morphism, then $\mathbf{F}(g)$ makes this diagram commutative for all $s \in (Y \times B)^A$

$$\begin{array}{ccc} & A & \\ s \swarrow & & \searrow \mathbf{F}(g)(s) \\ X \times B & \xrightarrow{g \times \text{id}_B} & Y \times B \end{array}$$

In particular, we have for $s := \mathbf{F}(f)(t)$ with an arbitrary $t \in (X \times B)^A$ this commutative diagram

$$\begin{array}{ccc} & A & \\ \mathbf{F}(f)(t) \swarrow & & \searrow \mathbf{F}(g)(\mathbf{F}(f)(t)) \\ Y \times B & \xrightarrow{g \times \text{id}_B} & Z \times B \end{array}$$

Thus the outer diagram commutes

$$\begin{array}{ccccc} & & A & & \\ & t \swarrow & \downarrow \mathbf{F}(f)(t) & \searrow \mathbf{F}(g)(\mathbf{F}(f)(t)) & \\ X \times B & \xrightarrow{f \times \text{id}_B} & Y \times B & \xrightarrow{g \times \text{id}_B} & Z \times B \end{array}$$

Consequently, we have

$$\begin{aligned} \mathbf{F}(g)(\mathbf{F}(f)(t)) &= (g \times \text{id}_B) \circ (f \times \text{id}_B) \circ t \\ &= ((g \circ f) \times \text{id}_B) \circ t \\ &= \mathbf{F}(g \circ f)(t). \end{aligned}$$

Now $\mathbf{F}(\text{id}_X) = \text{id}_{(X \times B)^A}$ is trivial, so that we have established indeed that $\mathbf{F} : \mathbf{AutO} \rightarrow \mathbf{Set}$ is a functor, assigning states to possible state transitions. \mathfrak{H}

Example 1.68 A *labeled transition system* is a collection of transitions indexed by a set of actions. Formally, given a set A of actions, $(S, (\rightsquigarrow_a)_{a \in A})$ is a labeled transition system iff $\rightsquigarrow_a \subseteq S \times S$ for all $a \in A$. Thus state s may go into state s' after action $a \in A$; this is written as $s \rightsquigarrow_a s'$. A morphism $f : (S, (\rightsquigarrow_{S,a})_{a \in A}) \rightarrow (T, (\rightsquigarrow_{T,a})_{a \in A})$ of transition systems is a map $f : S \rightarrow T$ such that $s \rightsquigarrow_{S,a} s'$ implies $f(s) \rightsquigarrow_{T,a} f(s')$ for all actions a , cp. Example 1.9.

We model a transition system $(S, (\rightsquigarrow_a)_{a \in A})$ as a map $F : S \rightarrow \mathcal{P}(A \times S)$ with $F(s) := \{\langle a, s' \rangle \mid s \rightsquigarrow_a s'\}$ (or, conversely, $\rightsquigarrow_a = \{\langle s, s' \rangle \mid \langle a, s' \rangle \in F(s)\}$), thus $F(s) \subseteq A \times S$ collects actions and new states. This suggests defining a map $\mathbf{F}(S) := \mathcal{P}(A \times S)$ which can be made a functor once we have decided what to do with morphisms $f : (S, (\rightsquigarrow_{S,a})_{a \in A}) \rightarrow (T, (\rightsquigarrow_{T,a})_{a \in A})$. Take $V \subseteq A \times S$ and define $\mathbf{F}(f)(V) := \{\langle a, f(s) \rangle \mid \langle a, s \rangle \in V\}$, (clearly we want to leave the actions alone). Then we have

$$\begin{aligned} \mathbf{F}(g \circ f)(V) &= \{\langle a, g(f(s)) \rangle \mid \langle a, s \rangle \in V\} \\ &= \{\langle a, g(y) \rangle \mid \langle a, y \rangle \in \mathbf{F}(f)(V)\} \\ &= \mathbf{F}(g)(\mathbf{F}(f)(V)) \end{aligned}$$

for a morphism $g : (T, (\rightsquigarrow_{T,a})_{a \in A}) \rightarrow (U, (\rightsquigarrow_{U,a})_{a \in A})$. Thus we have shown that $\mathbf{F}(g \circ f) = \mathbf{F}(g) \circ \mathbf{F}(f)$ holds. Because \mathbf{F} maps the identity to the identity, \mathbf{F} is a functor from the category of labeled transition systems to \mathbf{Set} . \mathfrak{H}

The next examples deal with functors induced by probabilities.

Example 1.69 Given a set X , define the support $\text{supp}(p)$ for a map $p : X \rightarrow [0, 1]$ as $\text{supp}(p) := \{x \in X \mid p(x) \neq 0\}$. A *discrete probability* p on X is a map $p : X \rightarrow [0, 1]$

with finite support such that

$$\sum_{x \in X} p(x) := \sup \left\{ \sum_{x \in F} p(x) \mid F \subseteq \text{supp}(p) \right\} = 1.$$

Denote by

$$\mathbf{D}(X) := \{p : X \rightarrow [0, 1] \mid p \text{ is a discrete probability}\}$$

the set of all discrete probabilities. Let $f : X \rightarrow Y$ be a map, and define

$$\mathbf{D}(f)(p)(y) := \sum_{\{x \in X \mid f(x)=y\}} p(x).$$

Because $\mathbf{D}(f)(p)(y) > 0$ iff $y \in f[\text{supp}(p)]$, $\mathbf{D}(f)(p) : Y \rightarrow [0, 1]$ has finite support, and

$$\sum_{y \in Y} \mathbf{D}(f)(p)(y) = \sum_{y \in Y} \sum_{\{x \in X \mid f(x)=y\}} p(x) = \sum_{x \in X} p(x) = 1.$$

It is clear that $\mathbf{D}(\text{id}_X)(p) = p$, so we have to check whether $\mathbf{D}(g \circ f) = \mathbf{D}(g) \circ \mathbf{D}(f)$ holds.

We use a little trick for this, which will turn out to be helpful later as well. Define

$$p(A) := \sup \left\{ \sum_{x \in F} p(x) \mid F \subseteq A \cap \text{supp}(p) \right\}$$

for $p \in \mathbf{D}(X)$ and $A \subseteq X$, then p is a probability measure on $\mathcal{P}X$. Then $\mathbf{D}(f)(p)(y) = p(f^{-1}[\{y\}])$, and $\mathbf{D}(f)(B) = p(f^{-1}[B])$ for $B \subseteq Y$. Thus we obtain for the maps $f : X \rightarrow Y$ and $g : Y \rightarrow Z$

$$\begin{aligned} \mathbf{D}(g \circ f)(p)(z) &= p((g \circ f)^{-1}[\{z\}]) \\ &= p(f^{-1}[g^{-1}[\{z\}]]) \\ &= \mathbf{D}(f)(p)(g^{-1}[\{z\}]) \\ &= \mathbf{D}(f)(\mathbf{D}(g)(p))(z) \end{aligned}$$

Thus $\mathbf{D}(g \circ f) = \mathbf{D}(g) \circ \mathbf{D}(f)$, as claimed.

Hence \mathbf{D} is an endofunctor on **Set**, the *discrete probability functor*. It is immediate that all the arguments above hold also for probabilities the support of which is countable; but since we will discuss an interesting example on page 63 which deal with the finite case, we stick to that here. 🙅

There is a continuous version of this functor as well. We generalize things a bit and formulate the example for subprobabilities.

Example 1.70 We are now working in the category **Meas** of measurable spaces with measurable maps as morphisms. Given a measurable space (X, \mathcal{A}) , the set $\mathbb{S}(X, \mathcal{A})$ of all subprobability measures is a measurable space with the weak σ -algebra $w(\mathcal{A})$ associated with \mathcal{A} , see Example 1.14. Hence \mathbb{S} maps measurable spaces to measurable spaces. Define for a morphism $f : (X, \mathcal{A}) \rightarrow (Y, \mathcal{B})$

$$\mathbb{S}(f)(\mu)(B) := \mu(f^{-1}[B])$$

for $B \in \mathcal{B}$. Then $\mathbb{S}(f) : \mathbb{S}(X, \mathcal{A}) \rightarrow \mathbb{S}(Y, \mathcal{B})$ is $w(\mathcal{A})$ - $w(\mathcal{B})$ -measurable. Now let $g : (Y, \mathcal{B}) \rightarrow (Z, \mathcal{C})$ be a morphism in **Meas**, then we show as in Example 1.69 that

$$\mathbb{S}(g \circ f)(\mu)(C) = \mu(f^{-1} [g^{-1} [C]]) = \mathbb{S}(f)(\mathbb{S}(g)(\mu))(C),$$

for $C \in \mathcal{C}$, thus $\mathbb{S}(g \circ f) = \mathbb{S}(g) \circ \mathbb{S}(f)$. Since \mathbb{S} preserves the identity, $\mathbb{S} : \mathbf{Meas} \rightarrow \mathbf{Meas}$ is an endofunctor, the (continuous space) *probability functor*. \mathbb{M}

The next two examples deal with upper closed sets, the first one with these sets proper, the second one with a more refined version, viz., with ultrafilters. Upper closed sets are used, e.g., for the interpretation of game logic, a variant of modal logics, see Example 1.190.

Example 1.71 Call a subset $V \subseteq \mathcal{PS}$ *upper closed* iff $A \in V$ and $A \subseteq B$ together imply $B \in V$; for example, each filter is upper closed. Denote by

$$\mathbf{VS} := \{V \subseteq \mathcal{PS} \mid V \text{ is upper closed}\}$$

the set of all upper closed subsets of \mathcal{PS} . Given $f : S \rightarrow T$, define

$$(\mathbf{V}f)(V) := \{W \subseteq \mathcal{PT} \mid f^{-1} [W] \in V\}$$

for $V \in \mathbf{VS}$. Let $W \in \mathbf{V}(V)$ and $W_0 \supseteq W$, then $f^{-1} [W] \subseteq f^{-1} [W_0]$, so that $f^{-1} [W_0] \in V$, hence $\mathbf{V}f : \mathbf{VS} \rightarrow \mathbf{VS}$. It is easy to see that $\mathbf{V}(g \circ f) = \mathbf{V}(g) \circ \mathbf{V}(f)$, provided $f : S \rightarrow T$, and $g : T \rightarrow V$. Moreover, $\mathbf{V}(\text{id}_S) = \text{id}_{\mathbf{V}(S)}$. Hence \mathbf{V} is an endofunctor on the category **Set** of sets with maps as morphisms. \mathbb{M}

Ultrafilters are upper closed, but are much more complex than plain upper closed sets, since they are filters, and they are maximal. Thus we have to look a bit closer at the properties which the functor is to represent.

Example 1.72 Let

$$\mathbf{US} := \{q \mid q \text{ is an ultrafilter over } S\}$$

assign to each set S its ultrafilters, to be more precise, all ultrafilters of the power set of S . This is the object part of an endofunctor over the category **Set** with maps as morphisms. Given a map $f : S \rightarrow T$, we have to define $\mathbf{U}f : \mathbf{US} \rightarrow \mathbf{UT}$. Before doing so, a preliminary consideration will help.

One first notes that, given two Boolean algebras B and B' and a Boolean algebra morphism $\gamma : B \rightarrow B'$, γ^{-1} maps ultrafilters over B' to ultrafilters over B . In fact, let w be an ultrafilter over B' , put $v := \gamma^{-1} [w]$; we go quickly over the properties of an ultrafilter should have. First, v does not contain the bottom element \perp_B of B , for otherwise $\perp_{B'} = \gamma(\perp_B) \in w$. If $a \in v$ and $b \geq a$, then $\gamma(b) \geq \gamma(a) \in w$, hence $\gamma(b) \in w$, thus $b \in v$; plainly, v is closed under \wedge . Now assume $a \notin v$, then $\gamma(a) \notin w$, hence $\gamma(-a) = -\gamma(a) \in w$, since w is an ultrafilter. Consequently, $-a \in v$. This establishes the claim.

Given a map $f : S \rightarrow T$, define $F_f : \mathcal{PT} \rightarrow \mathcal{PS}$ through $F_f := f^{-1}$. This is a homomorphism of the Boolean algebras \mathcal{PT} and \mathcal{PS} , thus F_f^{-1} maps \mathbf{US} to \mathbf{UT} . Put $\mathbf{U}(f) := F_f^{-1}$; note that we reverse the arrows' directions twice. It is clear that $\mathbf{U}(\text{id}_S) = \text{id}_{\mathbf{U}(S)}$, and if $g : T \rightarrow Z$, then

$$\mathbf{U}(g \circ f) = F_{g \circ f}^{-1} = (F_g \circ F_f)^{-1} = F_g^{-1} \circ F_f^{-1} = \mathbf{U}(g) \circ \mathbf{U}(f).$$

This shows that \mathbf{U} is an endofunctor on the category **Set** of sets with maps as morphisms (\mathbf{U} is sometimes denoted by β). \mathbb{M}

We can use functors for constructing new categories from given ones. As an example we define the comma category associated with two functors.

Definition 1.73 Let $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$ and $\mathbf{G} : \mathbf{M} \rightarrow \mathbf{L}$ be functors. The comma category (\mathbf{F}, \mathbf{G}) associated with \mathbf{F} and \mathbf{G} has as objects the triplets $\langle \mathbf{a}, f, \mathbf{b} \rangle$ with objects \mathbf{a} from \mathbf{K} , \mathbf{b} from \mathbf{M} , and morphisms $f : \mathbf{F}\mathbf{a} \rightarrow \mathbf{G}\mathbf{b}$. A morphism $(\varphi, \psi) : \langle \mathbf{a}, f, \mathbf{b} \rangle \rightarrow \langle \mathbf{a}', f', \mathbf{b}' \rangle$ is a pair of morphisms $\varphi : \mathbf{a} \rightarrow \mathbf{a}'$ of \mathbf{K} and $\psi : \mathbf{b} \rightarrow \mathbf{b}'$ of \mathbf{M} such that this diagram commutes

$$\begin{array}{ccc} \mathbf{F}\mathbf{a} & \xrightarrow{\mathbf{F}\varphi} & \mathbf{F}\mathbf{a}' \\ f \downarrow & & \downarrow f' \\ \mathbf{G}\mathbf{b} & \xrightarrow{\mathbf{G}\psi} & \mathbf{G}\mathbf{b}' \end{array}$$

Composition of morphism is component wise.

The slice category \mathbf{K}/\mathbf{x} defined in Example 1.16 is apparently the comma category $(\text{Id}_{\mathbf{K}}, \Delta_{\mathbf{x}})$.

Functors can be composed, yielding a new functor. The proof for this statement is straightforward.

Proposition 1.74 Let $\mathbf{F} : \mathbf{C} \rightarrow \mathbf{D}$ and $\mathbf{G} : \mathbf{D} \rightarrow \mathbf{E}$ be functors. Define $(\mathbf{G} \circ \mathbf{F})\mathbf{a} := \mathbf{G}(\mathbf{F}\mathbf{a})$ for an object \mathbf{a} of \mathbf{C} , and $(\mathbf{G} \circ \mathbf{F})f := \mathbf{G}(\mathbf{F}f)$ for a morphism $f : \mathbf{a} \rightarrow \mathbf{b}$ in \mathbf{C} , then $\mathbf{G} \circ \mathbf{F} : \mathbf{C} \rightarrow \mathbf{E}$ is a functor. \dashv

1.3.2 Natural Transformations

We see that we can compose functors in an obvious way. This raises the question whether or not functors themselves form a category. But we do not yet have morphisms between functors at our disposal. Natural transformations will assume this rôle. Nevertheless, the question remains, but it will not be answered in the positive; this is so because morphisms between objects should form a set, and it will be clear that this is not the case. Pumplün [Pum99] points at some difficulties that might arise and arrives at the pragmatic view that for practical problems this question is not particularly relevant.

But let us introduce natural transformations between functors \mathbf{F}, \mathbf{G} now. The basic idea is that for each object \mathbf{a} , $\mathbf{F}\mathbf{a}$ is transformed into $\mathbf{G}\mathbf{a}$ in a way which is compatible with the structure of the participating categories.

Definition 1.75 Let $\mathbf{F}, \mathbf{G} : \mathbf{K} \rightarrow \mathbf{L}$ be covariant functors. A family $\eta = (\eta_{\mathbf{a}})_{\mathbf{a} \in |\mathbf{K}|}$ is called a natural transformation $\eta : \mathbf{F} \rightarrow \mathbf{G}$ iff $\eta_{\mathbf{a}} : \mathbf{F}\mathbf{a} \rightarrow \mathbf{G}\mathbf{a}$ is a morphism in \mathbf{L} for all objects \mathbf{a} in \mathbf{K} such that this diagram commutes for any morphism $f : \mathbf{a} \rightarrow \mathbf{b}$ in \mathbf{K}

$$\begin{array}{ccc} \mathbf{a} & & \mathbf{F}\mathbf{a} \xrightarrow{\eta_{\mathbf{a}}} \mathbf{G}\mathbf{a} \\ f \downarrow & & \downarrow \mathbf{F}f \quad \quad \downarrow \mathbf{G}f \\ \mathbf{b} & & \mathbf{F}\mathbf{b} \xrightarrow{\eta_{\mathbf{b}}} \mathbf{G}\mathbf{b} \end{array}$$

Thus a natural transformation $\eta : \mathbf{F} \rightarrow \mathbf{G}$ is a family of morphisms, indexed by the objects of the common domain of \mathbf{F} and \mathbf{G} ; $\eta_{\mathbf{a}}$ is called the *component of η at \mathbf{a}* .

If \mathbf{F} and \mathbf{G} are both contravariant functors $\mathbf{K} \rightarrow \mathbf{L}$, we may perceive them as covariant functors $\mathbf{K} \rightarrow \mathbf{L}^{\text{op}}$, so that we get for the contravariant case this diagram:

$$\begin{array}{ccccc} & & \mathbf{F}a & \xrightarrow{\eta_a} & \mathbf{G}a \\ & & \uparrow \mathbf{F}f & & \uparrow \mathbf{G}f \\ a & \xrightarrow{f} & b & & \\ & & \mathbf{F}b & \xrightarrow{\eta_b} & \mathbf{G}b \end{array}$$

Let us have a look at some examples.

Example 1.76 $a_+ : x \mapsto \text{hom}_{\mathbf{K}}(a, x)$ yields a (covariant) functor $\mathbf{K} \rightarrow \mathbf{Set}$ for each object a in \mathbf{K} , see Example 1.61 (just for simplifying notation, we use again a_+ rather than $\text{hom}_{\mathbf{K}}(a, -)$, see page 14). Let $\tau : b \rightarrow a$ be a morphism in \mathbf{K} , then this induces a natural transformation $\eta_\tau : a_+ \rightarrow b_+$ with

$$\eta_{\tau, x} : \begin{cases} a_+(x) & \rightarrow b_+(x) \\ g & \mapsto g \circ \tau \end{cases}$$

In fact, look at this diagram with a \mathbf{K} -morphism $f : x \rightarrow y$:

$$\begin{array}{ccccc} x & & a_+(x) & \xrightarrow{\eta_{\tau, x}} & b_+(x) \\ f \downarrow & & \downarrow a_+(f) & & \downarrow b_+(f) \\ y & & a_+(y) & \xrightarrow{\eta_{\tau, y}} & b_+(y) \end{array}$$

Then we have for $h \in a_+(x) = \text{hom}_{\mathbf{K}}(a, x)$

$$\begin{aligned} (\eta_{\tau, y} \circ a_+(f))(h) &= \eta_{\tau, y}(f \circ h) \\ &= (f \circ h) \circ \tau \\ &= f \circ (h \circ \tau) \\ &= b_+(f)(\eta_{\tau, x}(h)) \\ &= (b_+(f) \circ \eta_{\tau, x})(h) \end{aligned}$$

Hence η_τ is in fact a natural transformation. \mathcal{M}

This is an example in the category of groups:

Example 1.77 Let \mathbf{K} be the category of groups (see Example 1.7). It is not difficult to see that \mathbf{K} has products. Define for a group H the map $\mathbf{F}_H(G) := H \times G$ on objects, and if $f : G \rightarrow G'$ is a morphism in \mathbf{K} , define $\mathbf{F}_H(f) : H \times G \rightarrow H \times G'$ through $\mathbf{F}_H(f) : \langle h, g \rangle \mapsto \langle h, f(g) \rangle$. Then \mathbf{F}_H is an endofunctor on \mathbf{K} . Now let $\varphi : H \rightarrow K$ be a morphism. Then φ induces a natural transformation η_φ upon setting

$$\eta_{\varphi, G} : \begin{cases} \mathbf{F}_H & \rightarrow \mathbf{F}_K \\ \langle h, g \rangle & \mapsto \langle \varphi(h), g \rangle. \end{cases}$$

In fact, let $\psi : L \rightarrow L'$ be a group homomorphism, then this diagram commutes

$$\begin{array}{ccc}
L & & \mathbf{F}_H L \xrightarrow{\eta_{\varphi, L}} \mathbf{F}_K L \\
\psi \downarrow & & \downarrow \mathbf{F}_K \psi \\
L' & & \mathbf{F}_H L' \xrightarrow{\eta_{\varphi, L'}} \mathbf{F}_K L'
\end{array}$$

To see this, take $\langle h, \ell \rangle \in \mathbf{F}_H L = H \times L$, and chase it through the diagram:

$$(\eta_{\varphi, L'} \circ \mathbf{F}_H \psi)(h, \ell) = \langle \varphi(h), \varphi(\ell) \rangle = (\mathbf{F}_K(\psi) \circ \eta_{\varphi, L})(h, \ell).$$

☞

Consider as an example a comma category (\mathbf{F}, \mathbf{G}) (Definition 1.73). There are functors akin to a projection which permit to recover the original functors, and which are connected through a natural transformation. To be specific:

Proposition 1.78 *Let $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$ and $\mathbf{G} : \mathbf{M} \rightarrow \mathbf{L}$ be functors. Then there are functors $\mathbf{S} : (\mathbf{F}, \mathbf{G}) \rightarrow \mathbf{K}$ and $\mathbf{R} : (\mathbf{F}, \mathbf{G}) \rightarrow \mathbf{M}$ rendering this diagram commutative:*

$$\begin{array}{ccc}
(\mathbf{F}, \mathbf{G}) & \xrightarrow{\mathbf{R}} & \mathbf{M} \\
\mathbf{S} \downarrow & & \downarrow \mathbf{G} \\
\mathbf{K} & \xrightarrow{\mathbf{F}} & \mathbf{L}
\end{array}$$

There exists a natural transformation $\eta : \mathbf{F} \circ \mathbf{S} \rightarrow \mathbf{G} \circ \mathbf{R}$.

Proof Put for the object $\langle a, f, b \rangle$ of (\mathbf{F}, \mathbf{G}) and the morphism (φ, ψ)

$$\begin{aligned}
\mathbf{S}\langle a, f, b \rangle &:= a, & \mathbf{S}(\varphi, \psi) &:= \varphi, \\
\mathbf{R}\langle a, f, b \rangle &:= b, & \mathbf{R}(\varphi, \psi) &:= \psi.
\end{aligned}$$

Then it is clear that the desired equality holds. Moreover, $\eta_{\langle a, f, b \rangle} := f$ is the desired natural transformation. The crucial diagram commutes by the definition of morphisms in the comma category. \dashv

Example 1.79 Assume that the product $a \times b$ for the objects a and b in category \mathbf{K} exists, then Proposition 1.41 tells us that we have for each object d a bijection $p_d : \text{hom}_{\mathbf{K}}(d, a) \times \text{hom}_{\mathbf{K}}(d, b) \rightarrow \text{hom}_{\mathbf{K}}(d, a \times b)$. Thus $(\pi_a \circ p_d)(f, g) = f$ and $(\pi_b \circ p_d)(f, g) = g$ for every morphism $f : d \rightarrow a$ and $g : d \rightarrow b$. Actually, p_d is the component of a natural transformation $p : \mathbf{F} \rightarrow \mathbf{G}$ with $\mathbf{F} := \text{hom}_{\mathbf{K}}(-, a) \times \text{hom}_{\mathbf{K}}(-, b)$ and $\mathbf{G} := \text{hom}_{\mathbf{K}}(-, a \times b)$ (note that this is short hand for the obvious assignments to objects and functors). Both \mathbf{F} and \mathbf{G} are *contravariant* functors from \mathbf{K} to \mathbf{Set} . So in order to establish naturalness, we have to establish that the following diagram commutes

$$\begin{array}{ccc}
c & & \text{hom}_{\mathbf{K}}(c, a) \times \text{hom}_{\mathbf{K}}(c, b) \xrightarrow{p_c} \text{hom}_{\mathbf{K}}(c, a \times b) \\
f \downarrow & & \uparrow \mathbf{G} f \\
d & & \text{hom}_{\mathbf{K}}(d, a) \times \text{hom}_{\mathbf{K}}(d, b) \xrightarrow{p_d} \text{hom}_{\mathbf{K}}(d, a \times b)
\end{array}$$

Now take $\langle g, h \rangle \in \text{hom}_{\mathbf{K}}(\mathbf{d}, \mathbf{a}) \times \text{hom}_{\mathbf{K}}(\mathbf{d}, \mathbf{b})$, then

$$\begin{aligned}\pi_{\mathbf{a}}((\mathbf{p}_{\mathbf{c}} \circ \mathbf{F} f)(g, h)) &= g \circ f = \pi_{\mathbf{a}}((\mathbf{G} f) \circ \mathbf{p}_{\mathbf{d}})(g, h), \\ \pi_{\mathbf{b}}((\mathbf{p}_{\mathbf{c}} \circ \mathbf{F} f)(g, h)) &= h \circ f = \pi_{\mathbf{b}}((\mathbf{G} f) \circ \mathbf{p}_{\mathbf{d}})(g, h).\end{aligned}$$

From this, commutativity follows. \mathbb{M}

We will — for the sake of illustration — define two ways of composing natural transformations. One is somewhat canonical, since it is based on the composition of morphisms, the other one is a bit tricky, since it involves the functors directly. Let us have a look at the direct one first.

Lemma 1.80 *Let $\eta : \mathbf{F} \rightarrow \mathbf{G}$ and $\zeta : \mathbf{G} \rightarrow \mathbf{H}$ be natural transformations. Then $(\tau \circ \zeta)(\mathbf{a}) := \tau(\mathbf{a}) \circ \zeta(\mathbf{a})$ defines a natural transformation $\tau \circ \zeta : \mathbf{F} \rightarrow \mathbf{H}$.*

Proof Let \mathbf{K} be the domain of functor \mathbf{F} , and assume that $f : \mathbf{a} \rightarrow \mathbf{b}$ is a morphism in \mathbf{K} . Then we have this diagram

$$\begin{array}{ccccc} \mathbf{a} & & \mathbf{F}\mathbf{a} & \xrightarrow{\eta_{\mathbf{a}}} & \mathbf{G}\mathbf{a} & \xrightarrow{\zeta_{\mathbf{a}}} & \mathbf{H}\mathbf{a} \\ f \downarrow & & \mathbf{F}f \downarrow & & \mathbf{G}g \downarrow & & \downarrow \mathbf{H}f \\ \mathbf{b} & & \mathbf{F}\mathbf{b} & \xrightarrow{\eta_{\mathbf{b}}} & \mathbf{G}\mathbf{b} & \xrightarrow{\zeta_{\mathbf{b}}} & \mathbf{H}\mathbf{b} \end{array}$$

Then

$$\mathbf{H}(f) \circ (\tau \circ \zeta)_{\mathbf{a}} = \mathbf{H}(f) \circ \tau_{\mathbf{a}} \circ \zeta_{\mathbf{a}} = \tau_{\mathbf{b}} \circ \mathbf{G}(f) \circ \zeta_{\mathbf{a}} = \tau_{\mathbf{b}} \circ \zeta_{\mathbf{b}} \circ \mathbf{F}(f) = (\tau \circ \zeta)_{\mathbf{b}} \circ \mathbf{F}(f).$$

Hence the outer diagram commutes. \dashv

The next composition is slightly more involved.

Proposition 1.81 *Given natural transformations $\eta : \mathbf{F} \rightarrow \mathbf{G}$ and $\tau : \mathbf{S} \rightarrow \mathbf{R}$ for functors $\mathbf{F}, \mathbf{G} : \mathbf{K} \rightarrow \mathbf{L}$ and $\mathbf{S}, \mathbf{R} : \mathbf{L} \rightarrow \mathbf{M}$. Then $\tau_{\mathbf{G}\mathbf{a}} \circ \mathbf{S}(\eta_{\mathbf{a}}) = \mathbf{R}(\eta_{\mathbf{a}}) \circ \tau_{\mathbf{F}\mathbf{a}}$ always holds. Put $(\tau * \eta)_{\mathbf{a}} := \tau_{\mathbf{G}\mathbf{a}} \circ \mathbf{S}(\eta_{\mathbf{a}})$. Then $\tau * \eta$ defines a natural transformation $\mathbf{S} \circ \mathbf{F} \rightarrow \mathbf{R} \circ \mathbf{G}$. $\tau * \eta$ is called the Godement product of η and τ .*

Proof 1. Because $\eta_{\mathbf{a}} : \mathbf{F}\mathbf{a} \rightarrow \mathbf{G}\mathbf{a}$, this diagram commutes by naturality of τ :

$$\begin{array}{ccc} \mathbf{S}(\mathbf{F}\mathbf{a}) & \xrightarrow{\tau_{\mathbf{F}\mathbf{a}}} & \mathbf{R}(\mathbf{F}\mathbf{a}) \\ \mathbf{S}(\eta_{\mathbf{a}}) \downarrow & & \downarrow \mathbf{R}(\eta_{\mathbf{a}}) \\ \mathbf{S}(\mathbf{G}\mathbf{a}) & \xrightarrow{\tau_{\mathbf{G}\mathbf{a}}} & \mathbf{R}(\mathbf{G}\mathbf{a}) \end{array}$$

This establishes the first claim.

2. Now let $f : \mathbf{a} \rightarrow \mathbf{b}$ be a morphism in \mathbf{K} , then the outer diagram commutes, since \mathbf{S} is a functor, and since τ is a natural transformation.

$$\begin{array}{ccccc}
& & (\tau * \eta)_a & & \\
& \swarrow & \text{---} & \searrow & \\
S(\mathbf{F}a) & \xrightarrow{\mathbf{F}(\eta_a)} & S(\mathbf{G}a) & \xrightarrow{\tau_{\mathbf{G}a}} & \mathbf{R}(\mathbf{G}a) \\
\downarrow S(\mathbf{F}f) & & \downarrow S(\mathbf{G}f) & & \downarrow \mathbf{R}(\mathbf{G}f) \\
S(\mathbf{F}b) & \xrightarrow{\mathbf{F}(\eta_b)} & S(\mathbf{G}b) & \xrightarrow{\tau_{\mathbf{G}b}} & \mathbf{R}(\mathbf{G}b) \\
& \nwarrow & \text{---} & \nearrow & \\
& & (\tau * \eta)_b & &
\end{array}$$

Hence $\tau * \eta : \mathbf{S} \circ \mathbf{F} \rightarrow \mathbf{R} \circ \mathbf{G}$ is natural indeed. \dashv

In [Lan97], $\eta \circ \tau$ is called the *vertical*, and $\eta * \tau$ the *horizontal* composition of the natural transformations η and τ . If $\eta : \mathbf{F} \rightarrow \mathbf{G}$ is a natural transformation, then the morphisms $(\mathbf{F}\eta)(a) := \mathbf{F}\eta_a : (\mathbf{F} \circ \mathbf{F})(a) \rightarrow (\mathbf{F} \circ \mathbf{G})(a)$ and $(\eta\mathbf{F})(a) := \eta_{\mathbf{F}a} : (\mathbf{F} \circ \mathbf{F})(a) \rightarrow (\mathbf{G} \circ \mathbf{F})(a)$ are available.

We know from Example 1.61 that $\text{hom}_{\mathbf{K}}(a, -)$ defines a covariant set valued functor; suppose we have another set valued functor $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{Set}$. Can we somehow compare these functors? This question looks on first sight quite strange, because we do not have any yardstick to compare these functors against. On second thought, we might use natural transformations for such an endeavor. It turns out that for any object a of \mathbf{K} the set $\mathbf{F}a$ is essentially given by the natural transformations $\eta : \text{hom}_{\mathbf{K}}(a, -) \rightarrow \mathbf{F}$. We will show now that there exists a bijective assignment between $\mathbf{F}a$ and these natural transformations. The reader might wonder about this somewhat intricate formulation; it is due to the observation that these natural transformation in general do not form a set but rather a class, so that we cannot set up a proper bijection (which would require sets as the basic scenario).

Lemma 1.82 *Let $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{Set}$ be a functor; given the object a of \mathbf{K} and a natural transformation $\eta : \text{hom}_{\mathbf{K}}(a, -) \rightarrow \mathbf{F}$, define the Yoneda isomorphism*

$$y_{a,\mathbf{F}}(\eta) := \eta(a)(\text{id}_a) \in \mathbf{F}a$$

Then $y_{a,\mathbf{F}}$ is bijective (i.e., onto, and one-to-one).

Proof 0. The assertion is established by defining for each $t \in \mathbf{F}a$ a natural transformation $\sigma_{a,\mathbf{F}}(t) : \text{hom}_{\mathbf{K}}(a, -) \rightarrow \mathbf{F}$ which is inverse to $y_{a,\mathbf{F}}$.

1. Given an object b of \mathbf{K} and $t \in \mathbf{F}a$, put

$$(\sigma_{a,\mathbf{F}}(t))_b := \sigma_{a,\mathbf{F}}(t)(b) : \begin{cases} \text{hom}_{\mathbf{K}}(a, b) & \rightarrow \mathbf{F}b \\ f & \mapsto (\mathbf{F}f)(t) \end{cases}$$

(note that $\mathbf{F}f : \mathbf{F}a \rightarrow \mathbf{F}b$ for $f : a \rightarrow b$, hence $(\mathbf{F}f)(t) \in \mathbf{F}b$). This defines a natural transformation $\sigma_{a,\mathbf{F}}(t) : \text{hom}_{\mathbf{K}}(a, -) \rightarrow \mathbf{F}$. In fact, if $f : b \rightarrow b'$, then

$$\begin{aligned}
\sigma_{a,\mathbf{F}}(t)(b')(\text{hom}_{\mathbf{K}}(a, f)g) &= \sigma_{a,\mathbf{F}}(t)(b')(f \circ g) \\
&= \mathbf{F}(f \circ g)(t) \\
&= (\mathbf{F}f)(\mathbf{F}g)(t) \\
&= (\mathbf{F}f)(\sigma_{a,\mathbf{F}}(t)(b)(g)).
\end{aligned}$$

Hence $\sigma_{a,F}(t)(b') \circ \text{hom}_{\mathbf{K}}(a, f) = (F f) \circ \sigma_{a,F}(t)(b)$.

2. We obtain

$$\begin{aligned} (y_{a,F} \circ \sigma_{a,F})(t) &= y_{a,F}(\sigma_{a,F}(t)) \\ &= \sigma_{a,F}(t)(a)(\text{id}_a) \\ &= (F \text{id}_a)(t) \\ &= \text{id}_{Fa}(t) \\ &= t \end{aligned}$$

That's not too bad, so let us try to establish that $\sigma_{a,F} \circ y_{a,F}$ is the identity as well. Given a natural transformation $\eta : \text{hom}_{\mathbf{K}}(a, -) \rightarrow F$, we obtain

$$(\sigma_{a,F} \circ y_{a,F})(\eta) = \sigma_{a,F}(y_{a,F}(\eta)) = \sigma_{a,F}(\eta_a(\text{id}_a)).$$

Thus we have to evaluate $\sigma_{a,F}(\eta_a(\text{id}_a))$. Take an object b and a morphism $f : a \rightarrow b$, then

$$\begin{aligned} \sigma_{a,F}(\eta_a(\text{id}_a)(b)(f)) &= (Ff)(\eta_a(\text{id}_a)) \\ &= (F(f) \circ \eta_a)(\text{id}_a) \\ &= (\eta_b \circ \text{hom}_{\mathbf{K}}(a, f))(\text{id}_a) && (\eta \text{ is natural}) \\ &= \eta_b(f) && (\text{since } \text{hom}_{\mathbf{K}}(a, f) \circ \text{id}_a = f \circ \text{id}_a = f) \end{aligned}$$

Thus $\sigma_{a,F}(\eta_a(\text{id}_a)) = \eta$. Consequently we have shown that $y_{a,F}$ is left and right invertible, hence is a bijection. \dashv

Now consider the set valued functor $\text{hom}_{\mathbf{K}}(b, -)$, then the Yoneda embedding says that $\text{hom}_{\mathbf{K}}(b, a)$ can be mapped bijectively to the natural transformations from $\text{hom}_{\mathbf{K}}(a, -)$ to $\text{hom}_{\mathbf{K}}(b, -)$. This entails these natural transformations being essentially the morphisms $b \rightarrow a$, and, conversely, each morphism $b \rightarrow a$ yields a natural transformation $\text{hom}_{\mathbf{K}}(a, -) \rightarrow \text{hom}_{\mathbf{K}}(b, -)$. The following statement makes this observation precise.

Proposition 1.83 *Given a natural transformation $\eta : \text{hom}_{\mathbf{K}}(a, -) \rightarrow \text{hom}_{\mathbf{K}}(b, -)$, there exists a unique morphism $g : b \rightarrow a$ such that $\eta_c(h) = h \circ g$ for every object c and every morphism $h : a \rightarrow c$ (thus $\eta = \text{hom}_{\mathbf{K}}(g, -)$).*

Proof 0. Let $y := y_{a, \text{hom}_{\mathbf{K}}(a, -)}$ and $\sigma := \sigma_{a, \text{hom}_{\mathbf{K}}(a, -)}$. Then y is a bijection with $(y \circ \sigma)(\eta) = \eta$ and $(\sigma \circ y)(h) = h$.

1. Put $g := \eta_a(\text{id}_a)$, then $g \in \text{hom}_{\mathbf{K}}(b, a)$, since $\eta_a : \text{hom}_{\mathbf{K}}(a, a) \rightarrow \text{hom}_{\mathbf{K}}(b, a)$ and $\text{id}_a \in \text{hom}_{\mathbf{K}}(a, a)$. Now let $h \in \text{hom}_{\mathbf{K}}(a, c)$, then

$$\begin{aligned} \eta_c(h) &= \sigma(\eta_a(\text{id}_a))(c)(h) && (\text{since } \eta = y \circ \sigma) \\ &= \sigma(g)(c)(h) && (\text{Definition of } \sigma) \\ &= \text{hom}_{\mathbf{K}}(b, g)(h) && (\text{hom}_{\mathbf{K}}(b, -) \text{ is the target functor}) \\ &= h \circ g \end{aligned}$$

2. If $\eta = \text{hom}_{\mathbf{K}}(g, -)$, then $\eta_a(\text{id}_a) = \text{hom}_{\mathbf{K}}(g, \text{id}_a) = \text{id}_a \circ g = g$, so $g : b \rightarrow a$ is uniquely determined. \dashv

A final example comes from measurable spaces, dealing with the weak- σ -algebra. We have defined in Example 1.63 the contravariant functor which assigns to each measurable space its σ -algebra, and we have defined in Example 1.14 the weak σ -algebra on its set of probability measures together with a set of generators. We show that this set of generators yields a family of natural transformations between the two contravariant functors involved.

Example 1.84 The contravariant functor $\mathbf{B} : \mathbf{Meas} \rightarrow \mathbf{Set}$ assigns to each measurable space its σ -algebra, and to each measurable map its inverse. Denote by $\mathbf{W} := \mathbf{P} \circ \mathbf{B}$ the functor that assigns to each measurable space the weak σ -algebra on its probability measures; $\mathbf{W} : \mathbf{Meas} \rightarrow \mathbf{Set}$ is contravariant as well. Recall from Example 1.14 that the set

$$\beta_S(A, r) := \{\mu \in \mathbb{S}(S, \mathcal{A}) \mid \mu(A) \geq r\}$$

denotes the set of all probability measures which evaluate the measurable set A not smaller than a given r , and that the weak σ -algebra on $\mathbb{S}(S, \mathcal{A})$ is generated by all these sets. We claim that $\beta(\cdot, r)$ is a natural transformation $\mathbf{B} \rightarrow \mathbf{W}$. Thus we have to show that this diagram commutes

$$\begin{array}{ccc} (S, \mathcal{A}) & & \mathbf{B}(S, \mathcal{A}) \xrightarrow{\beta_S(\cdot, r)} \mathbf{W}(S, \mathcal{A}) \\ f \downarrow & & \uparrow \mathbf{B} f \quad \quad \quad \uparrow \mathbf{W} f \\ (T, \mathcal{B}) & & \mathbf{B}(T, \mathcal{B}) \xrightarrow{\beta_T(\cdot, r)} \mathbf{W}(T, \mathcal{B}) \end{array}$$

Recall that we have $\mathbf{B}(f)(C) = f^{-1}[C]$ for $C \in \mathcal{B}$, and that $\mathbf{W}(f)(D) = \mathbb{S}(f)^{-1}[D]$, if $D \subseteq \mathbb{S}(T, \mathcal{B})$ is measurable. Now, given $C \in \mathcal{B}$, by expanding definitions we obtain

$$\begin{aligned} \mu \in \mathbf{W}(f)(\beta_T(C, r)) &\Leftrightarrow \mu \in \mathbb{S}(f)^{-1}[\beta_T(C, r)] \\ &\Leftrightarrow \mathbb{S}(f)(\mu) \in \beta_T(C, r) \\ &\Leftrightarrow \mathbb{S}(f)(\mu)(C) \geq r \\ &\Leftrightarrow \mu(f^{-1}[C]) \geq r \\ &\Leftrightarrow \mu \in \beta_S(\mathbf{B}(f)(C), r) \end{aligned}$$

Thus the diagram commutes in fact, and we have established that the generators for the weak σ -algebra come from a natural transformation. \mathbb{M}

1.3.3 Limits and Colimits

We have defined some constructions which permit to build new objects in a category from given ones, e.g., the product from two objects or the pushout. Each time we had some universal condition which had to be satisfied.

We will discuss the general construction very briefly and refer the reader to [Lan97, BW99, Pum99], where they are studied in great detail.

Definition 1.85 Given a functor $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$, a cone on \mathbf{F} consists of an object c in \mathbf{L} and of a family of morphisms $p_d : c \rightarrow \mathbf{F}d$ in \mathbf{L} for each object d in \mathbf{K} such that $p_{d'} = (\mathbf{F}g) \circ p_d$ for each morphism $g : d \rightarrow d'$ in \mathbf{K} .

So a cone $(c, (p_d)_{d \in |\mathbf{K}|})$ on \mathbf{F} looks like, well, a cone:

$$\begin{array}{ccc}
 & c & \\
 p_d \swarrow & & \searrow p_{d'} \\
 \mathbf{F}d & \xrightarrow{\mathbf{F}g} & \mathbf{F}d'
 \end{array}$$

$d \text{ --- } \frac{\quad}{g} \text{ --- } \succ d'$

A limiting cone provides a factorization for each other cone, to be specific

Definition 1.86 Let $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$ be a functor. The cone $(c, (p_d)_{d \in |\mathbf{K}|})$ is a limit of \mathbf{F} iff for every cone $(e, (q_d)_{d \in |\mathbf{K}|})$ on \mathbf{F} there exists a unique morphism $f : e \rightarrow c$ such that $q_d = p_d \circ f$ for each object d in \mathbf{K} .

Thus we have locally this situation for each morphism $g : d \rightarrow d'$ in \mathbf{K} :

$$\begin{array}{ccc}
 & e & \\
 q_d \swarrow & \downarrow f & \searrow q_{d'} \\
 & c & \\
 p_d \swarrow & & \searrow p_{d'} \\
 \mathbf{F}d & \xrightarrow{\mathbf{F}g} & \mathbf{F}d'
 \end{array}$$

$d \text{ --- } \frac{\quad}{g} \text{ --- } \succ d'$

The unique factorization probably gives already a clue at the application of this concept. Let us look at some examples.

Example 1.87 Let $X := \{1, 2\}$ and \mathbf{K} be the discrete category on X (see Example 1.6). Put $\mathbf{F}1 := a$ and $\mathbf{F}2 := b$ for the objects $a, b \in |\mathbf{L}|$. Assume that the product $a \times b$ with projections π_a and π_b exists in \mathbf{L} , and put $p_1 := \pi_a$, $p_2 := \pi_b$. Then $(a \times b, p_1, p_2)$ is a limit of \mathbf{F} . Clearly, this is a cone on \mathbf{F} , and if $q_1 : e \rightarrow a$ and $q_2 : e \rightarrow b$ are morphisms, there exists a unique morphism $f : e \rightarrow a \times b$ with $q_1 = p_1 \circ f$ and $q_2 = p_2 \circ f$ by the definition of a product. 🙌

The next example shows that a pullback can be interpreted as a limit.

Example 1.88 Let a, b, c objects in category \mathbf{L} with morphisms $f : a \rightarrow c$ and $g : b \rightarrow c$. Define category \mathbf{K} by $|\mathbf{K}| := \{a, b, c\}$, the hom-sets are defined as follows

$$\text{hom}_{\mathbf{K}}(x, y) := \begin{cases} \{\text{id}_x\}, & x = y \\ \{f\}, & x = a, y = c \\ \{g\}, & x = b, y = c \\ \emptyset, & \text{otherwise} \end{cases}$$

Let \mathbf{F} be the identity on $|\mathbf{K}|$ with $\mathbf{F}f := f$, $\mathbf{F}g := g$, and $\mathbf{F}\text{id}_x := \text{id}_x$ for $x \in |\mathbf{K}|$. If object p together with morphisms $t_a : p \rightarrow a$ and $t_b : p \rightarrow b$ is a pullback for f and g , then it is immediate that (p, t_a, t_b, t_c) is a limit cone for \mathbf{F} , where $t_c := f \circ t_a = g \circ t_b$. 🙌

Dualizing the concept of a cone, we obtain cocones.

Definition 1.89 *Given a functor $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$, an object $c \in |\mathbf{L}|$ together with morphisms $s_d : \mathbf{F}d \rightarrow c$ for each object d of \mathbf{K} such that $s_d = s_{d'} \circ \mathbf{F}g$ for each morphism $g : d \rightarrow d'$ is called a cocone on \mathbf{F} .*

Thus we have this situation

$$\begin{array}{ccc}
 d & \xrightarrow{\quad g \quad} & d' \\
 \mathbf{F}d & \xrightarrow{\quad \mathbf{F}g \quad} & \mathbf{F}d' \\
 \searrow s_d & & \swarrow s_{d'} \\
 & c &
 \end{array}$$

A colimit is then defined for a cocone.

Definition 1.90 *A cocone $(c, (s_d)_{d \in |\mathbf{K}|})$ is called a colimit for the functor $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$ iff for every cocone $(e, (t_d)_{d \in |\mathbf{K}|})$ for \mathbf{F} there exists a unique morphism $f : c \rightarrow e$ such that $t_d = f \circ s_d$ for every object $d \in |\mathbf{K}|$.*

So this yields

$$\begin{array}{ccc}
 d & \xrightarrow{\quad g \quad} & d' \\
 \mathbf{F}d & \xrightarrow{\quad \mathbf{F}g \quad} & \mathbf{F}d' \\
 \searrow s_d & & \swarrow s_{d'} \\
 & c & \\
 \swarrow t_d & \downarrow f & \searrow t_{d'} \\
 & e &
 \end{array}$$

Let us have a look at coproducts as an example.

Example 1.91 Let a and b be objects in category \mathbf{L} and assume that their coproduct $a + b$ with injections j_a and j_b exists in \mathbf{L} . Take again $I := \{1, 2\}$ and let \mathbf{K} be the discrete category over I . Put $\mathbf{F}1 := a$ and $\mathbf{F}2 := b$, then it follows from the definition of the coproduct that the cocone $(a + b, j_a, j_b)$ is a colimit for \mathbf{F} . ∇

One shows that the pushout can be represented as a colimit in the same way as in Example 1.88 for the representation of the pullback as a limit.

Both limits and colimits are powerful general concepts for representing important constructions with and on categories. We will encounter them later on, albeit mostly indirectly.

1.4 Monads and Kleisli Triples

We have now functors and natural transformations at our disposal, and we will put them to work. The first application we will tackle concerns monads. Moggi's work [Mog91, Mog89]

shows a connection between monads and computation which we will discuss now. Kleisli tripels as a practical disguise for monads are introduced first, and it will be shown through Manes' Theorem that they are equivalent in the sense that each Kleisli tripel generates a monad, and vice versa in a reversible construction. Some examples for monads follow, and we will finally have a brief look at the monadic construction in the programming language Haskell.

1.4.1 Kleisli Tripels

Assume that we work in a category \mathbf{K} and interpret values and computations of a programming language in \mathbf{K} . We need to distinguish between the values of a type \mathbf{a} and the computations of type \mathbf{a} , which are of type $\mathbf{T}\mathbf{a}$. For example

Non-deterministic computations Taking the values from a set A yields computations of type $\mathbf{T}A = \mathcal{P}_f(A)$, where the latter denotes all finite subsets of A .

Probabilistic computations Taking values from a set A will give computations in the set $\mathbf{T}A = \mathbf{D}A$ of all discrete probabilities on A , see Example 1.69.

Exceptions Here values of type A will result in values taken from $\mathbf{T}A = A + E$ with E as the set of *exceptions*.

Side effects Let L is the set of addresses in the store and U the set of all storage cells, a computation of type A will assign each element of U^L an element of A or another element of U^L , thus we have $\mathbf{T}A = (A + U^L)^{U^L}$.

Interactive input Let U be the set of characters, then $\mathbf{T}A$ is the set of all trees with finite fan out, so that the internal nodes have labels from U , and the leaves have labels taken from A .

In order to model this, we require an embedding of the values taken from \mathbf{a} into the computations of type $\mathbf{T}\mathbf{a}$, which is represented as a morphism $\eta_{\mathbf{a}} : \mathbf{a} \rightarrow \mathbf{T}\mathbf{a}$. Moreover, we want to be able to “lift” values to computations in this sense: if $f : \mathbf{a} \rightarrow \mathbf{T}\mathbf{b}$ is a map from values to computations, we want to extend f to a map $f^* : \mathbf{T}\mathbf{a} \rightarrow \mathbf{T}\mathbf{b}$ from computations to computations (thus we will be able to combine computations in a modular fashion). Understanding a morphism $\mathbf{a} \rightarrow \mathbf{T}\mathbf{b}$ as a program performing computations of type \mathbf{b} on values of type \mathbf{a} , this lifting will then permit performing computations of type \mathbf{b} depending on computations of type \mathbf{a} .

This leads to the definition of a Kleisli tripel.

Definition 1.92 Let \mathbf{K} be a category. A Kleisli tripel $(\mathbf{T}, \eta, -^*)$ over \mathbf{K} consists of a map $\mathbf{T} : |\mathbf{K}| \rightarrow |\mathbf{K}|$ on objects, a morphism $\eta_{\mathbf{a}} : \mathbf{a} \rightarrow \mathbf{T}\mathbf{a}$ for each object \mathbf{a} , an operation $*$ such that $f^* : \mathbf{T}\mathbf{a} \rightarrow \mathbf{T}\mathbf{b}$, if $f : \mathbf{a} \rightarrow \mathbf{T}\mathbf{b}$ with the following properties:

- ① $\eta_{\mathbf{a}}^* = \text{id}_{\mathbf{T}\mathbf{a}}$.
- ② $f^* \circ \eta_{\mathbf{a}} = f$, provided $f : \mathbf{a} \rightarrow \mathbf{T}\mathbf{b}$.
- ③ $g^* \circ f^* = (g^* \circ f)^*$ for $f : \mathbf{a} \rightarrow \mathbf{T}\mathbf{b}$ and $g : \mathbf{b} \rightarrow \mathbf{T}\mathbf{c}$.

Let us discuss briefly these properties of a Kleisli triple. The first property says that lifting the embedding $\eta_a : a \rightarrow Ta$ will give the identity on Ta . The second condition says that applying the lifted morphism f^* to an embedded value η_a will yield the same value as the given f . The third condition says that combining lifted morphisms is the same as lifting the lifted second morphism applied to the value of the first morphism.

The category associated with a Kleisli triple has the same objects as the originally given category (which is not too much of a surprise), but morphisms will correspond to programs: a program which performs a computation of type b on values of type a . Hence a morphism in this new category is of type $a \rightarrow Tb$ in the given one.

Definition 1.93 *Given a Kleisli triple $(T, \eta, -^*)$ over category K , the Kleisli category K_T is defined as follows*

- $|K_T| = |K|$, thus K_T has the same objects as K .
- $\text{hom}_{K_T}(a, b) = \text{hom}_K(a, Tb)$, hence f is a morphism $a \rightarrow b$ in K_T iff $f : a \rightarrow Tb$ is a morphism in K .
- The identity for a in K_T is $\eta_a : a \rightarrow Ta$.
- The composition $g * f$ of $f \in \text{hom}_{K_T}(a, b)$ and $g \in \text{hom}_{K_T}(b, c)$ is defined through $g * f := g^* \circ f$.

We have to show that Kleisli composition is associative: in fact, we have

$$\begin{aligned}
 (h * g) * f &= (h * g)^* \circ f \\
 &= (h^* \circ g)^* \circ f && \text{(definition of } h * g) \\
 &= h^* \circ g^* \circ f && \text{(property ③)} \\
 &= h^* \circ (g * f) && \text{(definition of } g * f) \\
 &= h * (f * g)
 \end{aligned}$$

Thus K_T is in fact a category. The map on objects in a Kleisli category extends to a functor (note that we did not postulate for a Kleisli triple that Tf is defined for morphisms). This functor is associated with two natural transformations which together form a monad. We will first define what a monad formally is, and then discuss the construction in some detail.

1.4.2 Monads

Definition 1.94 *A monad over a category K is a triple (T, η, μ) with these properties:*

- ❶ T is an endofunctor on K .
- ❷ $\eta : \text{Id}_K \rightarrow T$ and $\mu : T^2 \rightarrow T$ are natural transformations. η is called the unit, μ the multiplication of the monad.

③ *These diagrams commute*

$$\begin{array}{ccc}
 \mathbf{T}^3 \mathbf{a} & \xrightarrow{\mu_{\mathbf{T}\mathbf{a}}} & \mathbf{T}^2 \mathbf{a} \\
 \mathbf{T}\mu_{\mathbf{a}} \downarrow & & \downarrow \mu_{\mathbf{a}} \\
 \mathbf{T}^2 \mathbf{a} & \xrightarrow{\mu_{\mathbf{a}}} & \mathbf{T} \mathbf{a}
 \end{array}
 \qquad
 \begin{array}{ccccc}
 \mathbf{T} \mathbf{a} & \xrightarrow{\eta_{\mathbf{T}\mathbf{a}}} & \mathbf{T}^2 \mathbf{a} & \xleftarrow{\mathbf{T}\eta_{\mathbf{a}}} & \mathbf{T} \mathbf{a} \\
 & \searrow \text{id}_{\mathbf{T}\mathbf{a}} & \downarrow \mu_{\mathbf{a}} & \swarrow \text{id}_{\mathbf{T}\mathbf{a}} & \\
 & & \mathbf{T} \mathbf{a} & &
 \end{array}$$

Each Kleisli triple generates a monad, and vice versa. This is what Manes' Theorem says:

Theorem 1.95 *Given a category \mathbf{K} , there is a one-one correspondence between Kleisli triples and monads.*

Proof 1. Let $(\mathbf{T}, \eta, -^*)$ be a Kleisli triple. We will extend \mathbf{T} to a functor $\mathbf{K} \rightarrow \mathbf{K}$, and define the multiplication; the monad's unit will be η . Define

$$\begin{aligned}
 \mathbf{T}f &:= (\eta_{\mathbf{b}} \circ f)^*, \text{ if } f : \mathbf{a} \rightarrow \mathbf{b}, \\
 \mu_{\mathbf{a}} &:= (\text{id}_{\mathbf{T}\mathbf{a}})^*.
 \end{aligned}$$

Then μ is a natural transformation $\mathbf{T}^2 \rightarrow \mathbf{T}$. Clearly, $\mu_{\mathbf{a}} : \mathbf{T}^2 \mathbf{a} \rightarrow \mathbf{T} \mathbf{a}$ is a morphism. Let $f : \mathbf{a} \rightarrow \mathbf{b}$ be a morphism in \mathbf{K} , then we have

$$\begin{aligned}
 \mu_{\mathbf{b}} \circ \mathbf{T}^2 f &= \text{id}_{\mathbf{T}\mathbf{b}}^* \circ (\eta_{\mathbf{T}\mathbf{b}} \circ (\eta_{\mathbf{b}} \circ f)^*)^* \\
 &= (\text{id}_{\mathbf{T}\mathbf{b}}^* \circ (\eta_{\mathbf{T}\mathbf{b}} \circ (\eta_{\mathbf{b}} \circ f)^*))^* && \text{(by ③)} \\
 &= (\text{id}_{\mathbf{T}\mathbf{b}} \circ (\eta_{\mathbf{b}} \circ f)^*)^* && \text{(since } \text{id}_{\mathbf{T}\mathbf{b}}^* \circ \eta_{\mathbf{T}\mathbf{b}} = \text{id}_{\mathbf{T}\mathbf{b}}) \\
 &= (\eta_{\mathbf{b}} \circ f)^{**}
 \end{aligned}$$

Similarly, we obtain

$$(\mathbf{T}f) \circ \mu_{\mathbf{a}} = (\eta_{\mathbf{b}} \circ f)^* \circ \text{id}_{\mathbf{T}\mathbf{a}}^* = ((\eta_{\mathbf{b}} \circ f)^* \circ \text{id}_{\mathbf{T}\mathbf{a}})^* = (\eta_{\mathbf{b}} \circ f)^{**}.$$

Hence $\mu : \mathbf{T}^2 \rightarrow \mathbf{T}$ is natural. Because we obtain for the morphisms $f : \mathbf{a} \rightarrow \mathbf{b}$ and $g : \mathbf{b} \rightarrow \mathbf{c}$ the identity

$$(\mathbf{T}g) \circ (\mathbf{T}f) = (\eta_{\mathbf{c}} \circ g)^* \circ (\eta_{\mathbf{b}} \circ f)^* = ((\eta_{\mathbf{c}} \circ g)^* \circ \eta_{\mathbf{b}} \circ f)^* = (\eta_{\mathbf{c}} \circ g \circ f)^* = \mathbf{T}(g \circ f)$$

and since by ①

$$\mathbf{T} \text{id}_{\mathbf{a}} = (\eta_{\mathbf{a}} \circ \text{id}_{\mathbf{T}\mathbf{a}})^* = \eta_{\mathbf{a}}^* = \text{id}_{\mathbf{T}\mathbf{a}},$$

we conclude that \mathbf{T} is an endofunctor on \mathbf{K} .

We check the laws for unit and multiplication according to ③. One notes first that

$$\mu_{\mathbf{a}} \circ \eta_{\mathbf{T}\mathbf{a}} = \text{id}_{\mathbf{T}\mathbf{a}}^* \circ \eta_{\mathbf{T}\mathbf{a}} \stackrel{(\ddagger)}{=} \text{id}_{\mathbf{T}\mathbf{a}}$$

(in equation (\ddagger) we use ②), and that

$$\mu_{\mathbf{a}} \circ \mathbf{T}\mathbf{a} = \text{id}_{\mathbf{T}\mathbf{a}}^* (\eta_{\mathbf{T}\mathbf{a}} \circ \eta_{\mathbf{a}})^* = (\text{id}_{\mathbf{T}\mathbf{a}}^* \circ \eta_{\mathbf{T}\mathbf{a}} \circ \eta_{\mathbf{a}})^* \stackrel{(\ddagger)}{=} \eta_{\mathbf{a}}^* = (\eta_{\mathbf{a}} \circ \text{id}_{\mathbf{a}})^* = \mathbf{T}(\text{id}_{\mathbf{a}})$$

(in equation (†) we use ② again). Hence the rightmost diagram in ③ commutes. Turning to the leftmost diagram, we note that

$$\mu_a \circ \mu_{Ta} = \text{id}_{Ta}^* \circ \text{id}_{T^2a}^* = (\text{id}_{Ta}^* \circ \text{id}_{T^2a}^*)^* \stackrel{(\%)}{=} \mu_a^*,$$

using ③ in equation (%). On the other hand,

$$\mu_a \circ (\mathbf{T} \mu_a) = \text{id}_{Ta}^* \circ (\mathbf{T} \text{id}_{Ta}^*) = \text{id}_{Ta}^* \circ (\eta_{Ta} \circ \text{id}_{Ta}^*)^* = \text{id}_{Ta}^{**} = \mu_a^*,$$

because $\text{id}_{Ta}^* \circ \eta_{Ta} = \text{id}_{Ta}$ by ②. Hence the leftmost diagram commutes as well, and we have defined a monad indeed.

2. To establish the converse, define $f^* := \mu_b \circ (\mathbf{T}f)$ for the morphism $f : a \rightarrow Tb$. We obtain from the right hand triangle $\eta_a^* = \mu_a \circ (\mathbf{T}\eta_a) = \text{id}_{Ta}$, thus ① holds. Since $\eta : \text{Id}_K \rightarrow \mathbf{T}$ is natural, we have $(\mathbf{T}f) \circ \eta_a = \eta_{Tb} \circ f$ for $f : a \rightarrow Tb$. Hence

$$f^* \circ \eta_a = \mu_b \circ (\mathbf{T}f) \circ \eta_a = \mu_b \circ \eta_{Tb} \circ f = f$$

by the left hand side of the right triangle, giving ②. Finally, note that due to $\mu : \mathbf{T}^* \rightarrow \mathbf{T}$ being natural, we have for $g : b \rightarrow Tc$ the commutative diagram

$$\begin{array}{ccc} T^2b & \xrightarrow{\mu_b} & Tb \\ T^2g \downarrow & & \downarrow Tg \\ T^3c & \xrightarrow{\mu_{Tc}} & T^2c \end{array}$$

Then

$$\begin{aligned} g^* \circ f^* &= \mu_c \circ (\mathbf{T}g) \circ \mu_b \circ (\mathbf{T}f) \\ &= \mu_c \circ \mu_{Tc} \circ (T^2g) \circ (\mathbf{T}f) && (\text{since } (\mathbf{T}g) \circ \mu_b = \mu_{Tc} \circ T^2g) \\ &= \mu_c \circ (\mathbf{T}\mu_c) \circ \mathbf{T}(\mathbf{T}(g) \circ f) && (\text{since } \mu_c \circ \mu_{Tc} = \mu_c \circ (\mathbf{T}\mu_c)) \\ &= \mu_c \circ \mathbf{T}(\mu_c \circ \mathbf{T}(g) \circ f) \\ &= \mu_c \circ \mathbf{T}(g^* \circ f) \\ &= (g^* \circ f)^* \end{aligned}$$

This establishes ③ and shows that this defines a Kleisli triple. \dashv

Taking a Kleisli triple and producing a monad from it, one suspects that one might end up with a different Kleisli triple for the generated monad. But this is not the case; just for the record:

Corollary 1.96 *If the monad is given by a Kleisli triple, then the Kleisli triple defined by the monad coincides with the given one. Similarly, if the Kleisli triple is given by the monad, then the monad defined by the Kleisli triple coincides with the given one.*

Proof We use the notation from above. Given the monad, put $f^+ := \text{id}_{Tb} \circ (\eta_b \circ f)^*$, then

$$f^+ = \mu_{Tb} \circ (\eta_b \circ f)^* = (\text{id}_{Ta} \circ f)^* = f^*.$$

On the other hand, given the Kleisli triple, put $\mathbf{T}_0 f := (\eta_b \circ f)^*$, then

$$\mathbf{T}_0 f = \mu_b \circ \mathbf{T}(\eta_b \circ f) = \mu_b \circ \mathbf{T}(\eta_b) = \mathbf{T}f.$$

⊥

Let us have a look at some examples. Theorem 1.95 tells us that the specification of a Kleisli triple will give us the monad, and vice versa. Thus we are free to specify one or the other; usually the specification of the Kleisli triple is shorter and more concise.

Example 1.97 Nondeterministic computations may be modelled through a map $f : S \rightarrow \mathcal{P}(T)$: given a state (or an input, or whatever) from set S , the set $f(s)$ describes the set of all possible outcomes. Thus we work in category **Set** with maps as morphisms and take the power set functor \mathcal{P} as the functor. Define

$$\begin{aligned}\eta_S(x) &:= \{x\}, \\ f^*(B) &:= \bigcup_{x \in B} f(x)\end{aligned}$$

for the set S , for $B \subseteq S$ and the map $f : S \rightarrow \mathcal{P}(T)$. Then clearly $\eta_S : S \rightarrow \mathcal{P}(S)$, and $f^* : \mathcal{P}(S) \rightarrow \mathcal{P}(T)$. We check the laws for a Kleisli triple:

- ① Since $\eta_S^*(B) = \bigcup_{x \in B} \eta_S(x) = B$, we see that $\eta_S^* = \text{id}_{\mathcal{P}(S)}$
- ② It is clear that $f^* \circ \eta_a = f$ holds for $f : S \rightarrow \mathcal{P}(S)$.
- ③ Let $f : S \rightarrow \mathcal{P}(T)$ and $g : T \rightarrow \mathcal{P}(U)$, then

$$\begin{aligned}u \in (g^* \circ f^*)(B) &\Leftrightarrow u \in g(y) \text{ for some } x \in B \text{ and some } y \in f(x) \\ &\Leftrightarrow u \in g^*(f(x)) \text{ for some } x \in B\end{aligned}$$

$$\text{Thus } (g^* \circ f^*)(B) = (g^* \circ f)^*(B).$$

Hence the laws for a Kleisli triple are satisfied. Let us just compute $\mu_S = \text{id}_{\mathcal{P}(S)}^*$: Given $\beta \in \mathcal{P}(\mathcal{P}(S))$, we obtain

$$\mu_S(\beta) = \text{id}_{\mathcal{P}(S)}^*(\beta) = \bigcup_{B \in \beta} B = \bigcup \beta.$$

The same argumentation can be carried out when the power set functor is replaced by the finite power set functor $\mathcal{P}_f : S \mapsto \{A \subseteq S \mid A \text{ is finite}\}$ with the obvious definition of \mathcal{P}_f on maps. 🙌

In contrast to nondeterministic computations, probabilistic ones argue with probability distributions. We consider the discrete case first, and here we focus on probabilities with finite support.

Example 1.98 We work in the category **Set** of sets with maps as morphisms and consider the discrete probability functor $\mathbf{DS} := \{p : S \rightarrow [0, 1] \mid p \text{ is a discrete probability}\}$, see Example 1.69. Let $f : S \rightarrow \mathbf{DS}$ be a map and $p \in \mathbf{DS}$, put

$$f^*(p)(s) := \sum_{t \in S} f(t)(s)p(t).$$

Then

$$\sum_{s \in S} f^*(p)(s) = \sum_s \sum_t f(t)(s)p(t) = \sum_t \sum_s f(t)(s)p(t) = \sum_t p(t) = 1,$$

hence $f^* : \mathbf{DS} \rightarrow \mathbf{DS}$. Note that the set $\{\langle s, t \rangle \in S \times T \mid f(s)(t)p(s) > 0\}$ is finite, because p has finite support, and because each $f(s)$ has finite support as well. Since each of the summands is non-negative, we may reorder the summations at our convenience. Define moreover

$$\eta_S(s)(s') := d_S(s)(s') := \begin{cases} 1, & s = s' \\ 0, & \text{otherwise,} \end{cases}$$

so that $\eta_S(s)$ is the discrete *Dirac measure* on s . Then

① $\eta_S^*(p)(s) = \sum_{s'} d_S(s)(s')p(s') = p(s)$, hence we may conclude that $\eta_S^* \circ p = p$.

② $f^*(\eta_S)(s) = f(s)$ is immediate.

③ Let $f : S \rightarrow \mathbf{DT}$ and $g : T \rightarrow \mathbf{DU}$, then we have for $p \in \mathbf{DS}$ and $u \in U$

$$\begin{aligned} (g^* \circ f^*)(p)(u) &= \sum_{t \in T} g(t)(u)f^*(p)(t) \\ &= \sum_{t \in T} \sum_{s \in S} g(t)(u)f(s)(t)p(s) \\ &= \sum_{\langle s, t \rangle \in S \times T} g(t)(u)f(s)(t)p(s) \\ &= \sum_{s \in S} \left[\sum_{t \in T} g(t)(u)f(s)(t) \right] p(s) \\ &= \sum_{s \in S} g^*(f(s))(u)p(s) \\ &= (g^* \circ f)^*(p)(u) \end{aligned}$$

Again, we are not bound to any particular order of summation.

We obtain for $M \in (\mathbf{D} \circ \mathbf{D})S$

$$\mu_S(M)(s) = \text{id}_{\mathbf{DS}}^*(M)(s) = \sum_{q \in \mathbf{D}(S)} M(q) \cdot q(s).$$

The last sum extends over a finite set, because the support of M is finite. ✎

Since programs may fail to halt, one works sometimes in models which are formulated in terms of subprobabilities rather than probabilities. This is what we consider next, extending the previous example to the case of general measurable spaces.

Example 1.99 We work in the category of measurable spaces with measurable maps as morphisms, see Example 1.11. In Example 1.70 the subprobability functor was introduced, and it was shown that for a measurable space S the set $\mathbb{S}S$ of all subprobabilities is a measurable space again (we omit in this example the σ -algebra from notation, a measurable space is for the time being a pair consisting of a carrier set and a σ -algebra on it). A probabilistic computation f on the measurable spaces S and T produces from an input of an element of S

a subprobability distribution $f(s)$ on T , hence an element of $\mathbb{S}T$. We want f to be a morphism in **Meas**, so $f : S \rightarrow \mathbb{S}T$ is assumed to be measurable.

We know from Example 1.14 and Exercise 7 that $f : S \rightarrow \mathbb{S}T$ is measurable iff these conditions are satisfied:

1. $f(s) \in \mathbb{S}(T)$ for all $s \in S$, thus $f(s)$ is a subprobability on (the measurable sets of) T .
2. For each measurable set D in T , the map $s \mapsto f(s)(D)$ is measurable.

Returning to the definition of a Kleisli triple, we define for the measurable space S , $f : S \rightarrow \mathbb{S}T$,

$$\begin{aligned} e_S &:= \delta_S, \\ f^*(\mu)(B) &:= \int_S f(s)(B) \mu(ds) \quad (\mu \in \mathbb{S}S, B \subseteq T \text{ measurable}). \end{aligned}$$

Thus $e_S(x) = \delta_S(x)$, the Dirac measure associated with x , and $f^* : \mathbb{S}S \rightarrow \mathbb{S}S$ is a morphism (in this example, we write e for the unit, and m for the multiplication). Note that $f^*(\mu) \in \mathbf{S}(T)$ in the scenario above; in order to see whether the properties of a Kleisli triple are satisfied, we need to know how to integrate with this measure. Standard arguments show that

$$\int_T h \, df^*(\mu) = \int_S \int_T h(t) f(s)(dt) \mu(ds), \quad (1)$$

whenever $h : T \rightarrow \mathbb{R}_+$ is measurable and bounded.

Let us again check the properties a Kleisli triple. Fix B as a measurable subset of S , $f : S \rightarrow \mathbb{S}S$ and $g : T \rightarrow \mathbb{S}U$ as morphisms in **Meas**.

- ① Let $\mu \in \mathbb{S}S$, then

$$e_S^*(\mu)(B) = \int_S \delta_S(x)(B) \mu(dx) = \mu(B),$$

hence $e_S^* = \text{id}_{\mathbb{S}S}$.

- ② If $x \in S$, then

$$f^*(e_S(x))(B) = \int_S f(s)(B) \delta_S(x)(ds) = f(x)(B),$$

since $\int h \, d\delta_S(x) = h(x)$ for every measurable map h . Thus $f^* \circ e_S = f$.

- ③ Given $\mu \in \mathbb{S}S$, we have

$$\begin{aligned} (g^* \circ f^*)(\mu)(B) &= g^*(f^*(\mu))(B) \\ &= \int_T g(t)(B) f^*(\mu)(dt) \\ &\stackrel{(1)}{=} \int_S \int_T g(t)(B) f(s)(dt) \mu(ds) \\ &= \int_S g^*(f(s))(B) \mu(ds) \\ &= (g^* \circ f)^*(\mu)(B) \end{aligned}$$

Thus $g^* \circ f^* = (g^* \circ f)^*$.

Hence $(\mathbb{S}, \mathbf{e}, -^*)$ forms a Kleisli triple over the category **Meas** of measurable spaces.

Let us finally determine the monad's multiplication. We have for $M \in (\mathbb{S} \circ \mathbb{S})S$ and the measurable set $B \subseteq S$

$$m_S(M)(B) = \text{id}_{\mathbb{S}(S)}^*(M)(B) = \int_{\mathbb{S}(S)} \tau(B) M(d\tau)$$

✌

The underlying monad has been investigated by M. Giry, so it is called in her honor the *Giry monad*. It is used extensively as the machinery on which Markov transition systems are based.

The next example shows that ultrafilter define a monad as well.

Example 1.100 Let \mathbf{U} be the ultrafilter functor on **Set**, see Example 1.72. Define for the set S and the map $f : S \rightarrow \mathbf{U}T$

$$\begin{aligned} \eta_S(s) &:= \{A \subseteq S \mid s \in A\}, \\ f^*(U) &:= \{B \subseteq T \mid \{s \in S \mid B \in f(s)\} \in U\}, \end{aligned}$$

provided $U \in \mathbf{U}S$ is an ultrafilter. Then $\emptyset \notin f^*(U)$, since $\emptyset \notin U$. $\eta_S(s)$ is the principal ultrafilter associated with $s \in S$, hence $\eta_S : S \rightarrow \mathbf{U}S$. Because the intersection of two sets is a member of an ultrafilter iff both sets are elements of it,

$$\{s \in S \mid B_1 \cap B_2 \in f(s)\} = \{s \in S \mid B_1 \in f(s)\} \cap \{s \in S \mid B_2 \in f(s)\},$$

$f^*(U)$ is closed under intersections, moreover, $B \subseteq C$ and $B \in f^*(U)$ imply $C \in f^*(U)$. If $B \notin f^*(U)$, then $\{s \in S \mid f(s) \in B\} \notin U$, hence $\{s \in S \mid B \notin f(s)\} \in U$, thus $S \setminus B \in f^*(U)$, and vice versa. Hence $f^*(U)$ is an ultrafilter, thus $f^* : \mathbf{U}S \rightarrow \mathbf{U}T$.

We check whether $(\mathbf{U}, \eta, -^*)$ is a Kleisli triple.

- ① Since $B \in \eta_S^*$ iff $\{s \in S \mid s \in B\} \in U$, we conclude that $\eta_S^* = \text{id}_{\mathbf{U}S}$.
- ② Similarly, if $f : S \rightarrow \mathbf{U}T$ and $s \in S$, then $B \in (f^* \circ \eta_S)(s)$ iff $B \in f(s)$, hence $f^* \circ \eta_S = f$.
- ③ Let $f : S \rightarrow \mathbf{U}T$ and $g : T \rightarrow \mathbf{U}W$. Then

$$\begin{aligned} B \in (g^* \circ f^*)(U) &\Leftrightarrow \{s \in S \mid \{t \in T \mid B \in g(t)\} \in f(s)\} \in U \\ &\Leftrightarrow B \in (g^* \circ f)^*(U) \end{aligned}$$

for $U \in \mathbf{U}S$. Consequently, $g^* \circ f^* = (g^* \circ f)^*$.

Let us compute the monad's multiplication. Define for $B \subseteq S$ the set $[B] := \{C \in \mathbf{U}S \mid B \in C\}$ as the set of all ultrafilters on S which contain B as an element, then an easy computation shows

$$\mu_S(V) = \text{id}_{\mathbf{U}S}^*(V) = \{B \subseteq S \mid [B] \in V\}$$

for $V \in (\mathbf{U} \circ \mathbf{U})S$. ✌


Example 1.101 This example deals with upper closed subsets of the power set of a set, see Example 1.71. Let again

$$\mathbf{VS} := \{V \subseteq \mathcal{PS} \mid V \text{ is upper closed}\}$$

be the endofunctor on **Set** which assigns to set S all upper closed subsets of \mathcal{PS} . We define the components of a Kleisli triple as follows: $\eta_S(s)$ is the principal ultrafilter generated by $s \in S$, which is upper closed, and if $f : S \rightarrow \mathbf{VT}$ is a map, we put

$$f^*(V) := \{B \subseteq T \mid \{s \in S \mid B \in f(s)\} \in V\}$$

for $V \in \mathbf{VT}$, see in Example 1.100.

The argumentation in Example 1.100 carries over and shows that this defines a Kleisli triple. 

These examples show that monads and Kleisli tripels are constructions which model many computationally interesting subjects. After looking at the practical side of this, we return to the discussion of the relationship of monads to adjunctions, another important construction.

1.4.3 Monads in Haskell

The functional programming language **Haskell** thrives on the construction of monads. We have a brief look.

Haskell permits the definition of type classes; the definition of a type class requires the specification of the types on which the class is based, and the signature of the functions defined by this class. The definition of class **Monad** is given below (actually, it is rather a specification of Kleisli tripels).

```
class Monad m where
  (>>=)  :: m a -> (a -> m b) -> m b
  return :: a -> m a
  (>>)   :: m a -> m b -> m b
  fail   :: String -> m a
```

Thus class **Monad** is based on type constructor **m**, it specifies four functions of which **>>=** and **return** are the most interesting. The first one is called *bind* and used as an infix operator: given **x** of type **m a** and a function **f** of type **a -> m b**, the evaluation of **x >>= f** will yield a result of type **m b**. This corresponds to f^* . The function **return** takes a value of type **a** and evaluates to a value of type **m a**; hence it corresponds to η_a (the name **return** is probably not a fortunate choice). The function **>>**, usually used as an infix operator as well, is defined by default in terms of **>>=**, and function **fail** serves to handling exceptions; both functions will not concern us here.

Not every conceivable definition of the functions **return** and the bind function **>>=** are suitable for the definition of a monad. These are the laws the **Haskell** programmer has to enforce, and it becomes evident that these are just the laws for a Kleisli triple from Definition 1.92:

```

return x >>= f      == f x
p >>= return      == p
p >>= (\x -> (f x >>= g)) == (p >>= (\x -> f x)) >>= g

```

(here x is not free in g ; note that $\backslash x \rightarrow f\ x$ is Haskell's way of expressing the anonymous function $\lambda x.f x$). The compiler for Haskell cannot check these laws, so the programmer has to make sure that they hold.

We demonstrate the concept with a simple example. Lists are a popular data structure. They are declared as a monad in this way:

```

instance Monad [] where
    return t = [t]
    x >>= f  = concat (map f x)

```

This makes the polymorphic type constructor `[]` for lists a monad; it specifies essentially that f is mapped over the list x (note that x has to be a list, and f a function of the list's base type to a list); this yields a list of lists which then will be flattened through an application of function `concat`. This example should clarify things:

```

>>> q = (\w -> [0 .. w])
>>> [0 .. 2] >>= q
[0,0,1,0,1,2]

```

The effect is explained in the following way: The definition of the bind operation `>>=` requires the computation of

```

concat (map q [0 .. 2]) = concat [(q 0), (q 1), (q 2)]
= concat [[0], [0, 1], [0, 1, 2]]
= [0,0,1,0,1,2].

```

We check the laws of a monad.

- We have `return x == [x]`, hence

```

return x >>= f == [x] >>= f
               == concat (map f [x])
               == concat [f x]
               == f x

```

- Similarly, if p is a given list, then

```

p >>= return == concat (map return p)
              == concat [[x] | x <- p]
              == [x | x <- p]
              == p

```

- For the third law, if p is the empty list, then the left and the right hand side are empty as well. Hence let us assume that $p = [x_1, \dots, x_n]$. We obtain for the left hand side

```

p >>= (\x -> (f x >>= g))
== concat (map (\x -> (f x >>= g)) p)
== concat (concat [map g (f x) | x <- p]),

```

and for the right hand side

```

(concat [f x | x <- p]) >=> g
== ((f x1) ++ (f x2) ++ .. ++ (f xn)) >=> g
== concat (map g ((f x1) ++ (f x2) ++ .. ++ (f xn)))
== concat (concat [map g (f x) | x <- p])

```

(this argumentation could of course be made more precise through a proof by induction on the length of list p , but this would lead us too far from the present discussion).

Kleisli composition $>=>$ can be defined in a monad as follows:

```

(>=>) :: Monad m => (a -> m b) -> (b -> m c) -> (a -> m c)
f >=> g = \x -> (f x) >=> g

```

This gives in the first line a type declaration for operation $>=>$ by indicating that the infix operator $>=>$ takes two arguments, viz., a function the signature of which is $a \rightarrow m\ b$, a second one with signature $b \rightarrow m\ c$, and that the result will be a function of type $a \rightarrow m\ c$, as expected. The precondition to this type declaration is that m is a monad. The body of the function will use the bind operator for binding $f\ x$ to g ; this results in a function depending on x . It can be shown that this composition is associative.

1.5 Adjunctions and Algebras

Adjunctions. We define the basic notion of an adjunction and show that an adjunction defines a pair of natural transformations through universal arrows (which is sometimes taken as the basis for adjunctions).

Definition 1.102 *Let \mathbf{K} and \mathbf{L} be categories. Then $(\mathbf{F}, \mathbf{G}, \varphi)$ is called an adjunction iff*

1. $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$ and $\mathbf{G} : \mathbf{L} \rightarrow \mathbf{K}$ are functors,
2. for each object a in \mathbf{L} and x in \mathbf{K} there is a bijection

$$\varphi_{x,a} : \text{hom}_{\mathbf{L}}(\mathbf{F}x, a) \rightarrow \text{hom}_{\mathbf{K}}(x, \mathbf{G}a)$$

which is natural in x and a .

\mathbf{F} is called the left adjoint to \mathbf{G} , \mathbf{G} is called the right adjoint to \mathbf{F} .

That $\varphi_{x,a}$ is natural for each x, a means that for all morphisms $f : a \rightarrow b$ in \mathbf{L} and $g : y \rightarrow x$ in \mathbf{K} both diagrams commute:

$$\begin{array}{ccc}
 \text{hom}_{\mathbf{L}}(\mathbf{F}x, a) & \xrightarrow{\varphi_{x,a}} & \text{hom}_{\mathbf{K}}(x, \mathbf{G}a) \\
 f_* \downarrow & & \downarrow (\mathbf{G}f)_* \\
 \text{hom}_{\mathbf{L}}(\mathbf{F}x, b) & \xrightarrow{\varphi_{x,b}} & \text{hom}_{\mathbf{K}}(x, \mathbf{G}b)
 \end{array}
 \qquad
 \begin{array}{ccc}
 \text{hom}_{\mathbf{L}}(\mathbf{F}x, a) & \xrightarrow{\varphi_{x,a}} & \text{hom}_{\mathbf{K}}(x, \mathbf{G}a) \\
 (\mathbf{F}g)^* \downarrow & & \downarrow g^* \\
 \text{hom}_{\mathbf{L}}(\mathbf{F}y, a) & \xrightarrow{\varphi_{y,a}} & \text{hom}_{\mathbf{K}}(y, \mathbf{G}a)
 \end{array}$$

Here $f_* := \text{hom}_{\mathbf{L}}(\mathbf{F}x, f)$ and $g^* := \text{hom}_{\mathbf{K}}(g, \mathbf{G}a)$ are the hom-set functors associated with f resp. g , similar for $(\mathbf{G}f)_*$ and for $(\mathbf{F}g)^*$; for the hom-set functors see Example 1.61.

Let us have a look at a simple example: Currying.

Example 1.103 A map $f : X \times Y \rightarrow Z$ is sometimes considered as a map $f : X \rightarrow (Y \rightarrow Z)$, so that $f(x, y)$ is considered as the value at y for the map $f(x) := \lambda b. f(x, b)$. This helpful technique is called *currying* and will be discussed now.

Fix a set E and define the endofunctors $\mathbf{F}, \mathbf{G} : \mathbf{Set} \rightarrow \mathbf{Set}$ by $\mathbf{F} := - \times E$ resp. $\mathbf{G} := -^E$. Thus we have in particular $(\mathbf{F}f)(x, e) := \langle f(x), e \rangle$ and $(\mathbf{G}f)(g)(e) := f(g(e))$, whenever $f : X \rightarrow Y$ is a map.

Define the map $\varphi_{X,A} : \text{hom}_{\mathbf{Set}}(\mathbf{F}X, A) \rightarrow \text{hom}_{\mathbf{Set}}(X, \mathbf{G}A)$ by $\varphi_{X,A}(k)(x)(e) := k(x, e)$. Then $\varphi_{X,A}$ is a bijection. In fact, let $k_1, k_2 : \mathbf{F}X \rightarrow A$ be different maps, then $k_1(x, e) \neq k_2(x, e)$ for some $\langle x, e \rangle \in X \times E$, hence $\varphi_{X,A}(k_1)(x)(e) \neq \varphi_{X,A}(k_2)(x)(e)$, so that $\varphi_{X,A}$ is one-to-one. Let $\ell : X \rightarrow \mathbf{G}A$ be a map, then $\ell = \varphi_{X,A}(k)$ with $k(x, e) := \ell(x)(e)$. Thus $\varphi_{X,A}$ is onto.

In order to show that φ is natural both in X and in A , take maps $f : A \rightarrow B$ and $g : Y \rightarrow X$ and trace $k \in \text{hom}_{\mathbf{Set}}(\mathbf{F}X, A)$ through the diagrams in Definition 1.102. We have

$$\varphi_{X,B}(f_*(k))(x)(e) = f_*(k)(x, e) = f(k(x, e)) = f(\varphi_{X,A}(k)(x)(e)) = (\mathbf{G}f)_*(\varphi_{X,A}(k)(x)(e)).$$

Similarly,

$$g^*(\varphi_{X,A}(k))(y)(e) = k(g(y), e) = (\mathbf{F}g)^*(k)(y, e) = \varphi_{Y,A}((\mathbf{F}g)^*(k))(y)(e).$$

This shows that $(\mathbf{F}, \mathbf{G}, \varphi)$ with φ as the currying function is an adjunction. \mathcal{M}

Another popular example is furnished through the diagonal functor.

Example 1.104 Let \mathbf{K} be a category such that for any two objects a and b their product $a \times b$ exists. Recall the definition of the Cartesian product of categories from Lemma 1.19. Define the diagonal functor $\Delta : \mathbf{K} \rightarrow \mathbf{K} \times \mathbf{K}$ through $\Delta a := \langle a, a \rangle$ for objects and $\Delta f := \langle f, f \rangle$ for morphism f . Conversely, define $\mathbf{T} : \mathbf{K} \times \mathbf{K} \rightarrow \mathbf{K}$ by putting $\mathbf{T}(a, b) := a \times b$ for objects, and $\mathbf{T}\langle f, g \rangle := f \times g$ for morphism $\langle f, g \rangle$.

Let $\langle k_1, k_2 \rangle \in \text{hom}_{\mathbf{K} \times \mathbf{K}}(\Delta a, \langle b_1, b_2 \rangle)$, hence we have morphisms $k_1 : a \rightarrow b_1$ and $k_2 : a \rightarrow b_2$. By the definition of the product, there exists a unique morphism $k : a \rightarrow b_1 \times b_2$ with $k_1 = \pi_1 \circ k$ and $k_2 = \pi_2 \circ k$, where $\pi_i : b_1 \times b_2 \rightarrow b_i$ are the projections, $i = 1, 2$. Define $\varphi_{a, b_1 \times b_2}(k_1, k_2) := k$, then it is immediate that $\varphi_{a, b_1 \times b_2} : \text{hom}_{\mathbf{K} \times \mathbf{K}}(\Delta a, \langle b_1, b_2 \rangle) \rightarrow \text{hom}_{\mathbf{K}}(a, \mathbf{T}(b_1, b_2))$ is a bijection.

Let $\langle f_1, f_2 \rangle : \langle a_1, a_2 \rangle \rightarrow \langle b_1, b_2 \rangle$ be a morphism, then the diagram

$$\begin{array}{ccc} \text{hom}_{\mathbf{K} \times \mathbf{K}}(\Delta x, \langle a_1, a_2 \rangle) & \xrightarrow{\varphi_{x, \langle a_1, a_2 \rangle}} & \text{hom}_{\mathbf{K}}(x, a_1 \times a_2) \\ \langle f_1, f_2 \rangle_* \downarrow & & \downarrow (\mathbf{T}(f_1, f_2))_* \\ \text{hom}_{\mathbf{K} \times \mathbf{K}}(\Delta x, \langle b_1, b_2 \rangle) & \xrightarrow{\varphi_{x, \langle b_1, b_2 \rangle}} & \text{hom}_{\mathbf{K}}(x, b_1 \times b_2) \end{array}$$

splits into the two commutative diagrams

$$\begin{array}{ccc} \text{hom}_{\mathbf{K}}(x, a_i) & \xrightarrow{\pi_i \circ \varphi_{x, \langle a_1, a_2 \rangle}} & \text{hom}_{\mathbf{K}}(x, a_i) \\ f_{i,*} \downarrow & & \downarrow (\pi_i \circ (\mathbf{T}(f_1, f_2)))_* \\ \text{hom}_{\mathbf{K}}(x, b_i) & \xrightarrow{\pi_i \circ \varphi_{x, \langle b_1, b_2 \rangle}} & \text{hom}_{\mathbf{K}}(x, b_i) \end{array}$$

for $i = 1, 2$, hence is commutative itself. One argue similarly for a morphism $g : b \rightarrow a$. Thus the bijection φ is natural.

Hence we have found out that $(\Delta, \mathbf{T}, \varphi)$ is an adjunction, so that the diagonal functor has the product functor as an adjoint. \mathbb{N}

A map $f : X \rightarrow Y$ between sets provides us with another example, which is the special case of a Galois connection (recall that a pair $f : P \rightarrow Q$ and $g : Q \rightarrow P$ of monotone maps between the partially ordered sets P and Q form a *Galois connection* iff $f(p) \geq q \Leftrightarrow p \leq g(q)$ for all $p \in P, q \in Q$).

Example 1.105 Let X and Y be sets, then the inclusion on $\mathcal{P}X$ resp. $\mathcal{P}Y$ makes these sets categories, see Example 1.4. Given a map $f : X \rightarrow Y$, define $f_? : \mathcal{P}X \rightarrow \mathcal{P}Y$ as the direct image $f_?(A) := f[A]$ and $f_! : \mathcal{P}Y \rightarrow \mathcal{P}X$ as the inverse image $f_!(B) := f^{-1}[B]$. Now we have for $A \subseteq X$ and $B \subseteq Y$

$$B \subseteq f_?(A) \Leftrightarrow B \subseteq f[A] \Leftrightarrow f^{-1}[B] \subseteq A \Leftrightarrow f_!(B) \subseteq A.$$

This means in terms of the hom-sets that $\text{hom}_{\mathcal{P}Y}(B, f_?(A)) \neq \emptyset$ iff $\text{hom}_{\mathcal{P}X}(f_!(B), A) \neq \emptyset$. Hence this gives an adjunction $(f_!, f_?, \varphi)$. \mathbb{N}

Back to the general development. This auxiliary statement will help in some computations.

Lemma 1.106 Let $(\mathbf{F}, \mathbf{G}, \varphi)$ be an adjunction, $f : a \rightarrow b$ and $g : y \rightarrow x$ be morphisms in \mathbf{L} resp. \mathbf{K} . Then we have

$$\begin{aligned} (\mathbf{G}f) \circ \varphi_{x,a}(t) &= \varphi_{x,b}(f \circ t), \\ \varphi_{x,a}(t) \circ g &= \varphi_{y,a}(t \circ \mathbf{F}g) \end{aligned}$$

for each morphism $t : \mathbf{F}x \rightarrow a$ in \mathbf{L} .

Proof Chase t through the left hand diagram of Definition 1.102 to obtain

$$((\mathbf{G}f)_* \circ \varphi_{x,a})(t) = (\mathbf{G}f) \circ \varphi_{x,a}(t) = \varphi_{x,b}(f_*(t)) = \varphi_{x,b}(f \circ t).$$

This yields the first equation, the second is obtained from tracing t through the diagram on the right hand side. \dashv

An adjunction induces natural transformations which make this important construction easier to handle, and which helps indicating connections of adjunctions to monads and Eilenberg-Moore algebras in the sequel. Before entering the discussion, universal arrows are introduced.

Definition 1.107 Let $\mathbf{S} : \mathbf{C} \rightarrow \mathbf{D}$ be a functor, and c an object in \mathbf{C} .

1. the pair $\langle r, u \rangle$ is called a universal arrow from c to \mathbf{S} iff r is an object in \mathbf{C} and $u : c \rightarrow \mathbf{S}r$ is a morphism in \mathbf{D} such that for any arrow $f : c \rightarrow \mathbf{S}d$ there exists a unique arrow $f' : r \rightarrow d$ in \mathbf{C} such that $f = (\mathbf{S}f') \circ u$.
2. the pair $\langle r, v \rangle$ is called a universal arrow from \mathbf{S} to c iff r is an object in \mathbf{C} and $v : \mathbf{S}r \rightarrow c$ is a morphism in \mathbf{D} such that for any arrow $f : \mathbf{S}d \rightarrow c$ there exists a unique arrow $f' : d \rightarrow r$ in \mathbf{C} such that $f = v \circ (\mathbf{S}f')$.

Thus, if the pair $\langle r, u \rangle$ is universal from c to S , then each arrow $c \rightarrow Sd$ in C factors uniquely through the S -image of an arrow $r \rightarrow d$ in C . Similarly, if the pair $\langle r, v \rangle$ is universal from S to c , then each D -arrow $Sd \rightarrow c$ factors uniquely through the S -image of an C -arrow $d \rightarrow r$. These diagrams depict the situation for a universal arrow $u : c \rightarrow Sr$ resp. a universal arrow $v : Sr \rightarrow c$.

$$\begin{array}{ccc}
 c & \xrightarrow{u} & Sr \\
 & \searrow f & \downarrow Sf' \\
 & & Sd
 \end{array}
 \quad
 \begin{array}{ccc}
 r & \xrightarrow{f'} & d \\
 \vdots & & \vdots
 \end{array}
 \quad
 \begin{array}{ccc}
 Sr & \xrightarrow{v} & c \\
 \uparrow Sf' & \nearrow f & \\
 Sd & &
 \end{array}
 \quad
 \begin{array}{ccc}
 r & \xrightarrow{f'} & d \\
 \vdots & & \vdots
 \end{array}$$

This is a characterization of a universal arrow from c to S .

Lemma 1.108 *Let $S : C \rightarrow D$ be a functor. Then $\langle r, u \rangle$ is an universal arrow from c to S iff the function ψ_d which maps each morphism $f' : r \rightarrow d$ to the morphism $(Sf') \circ u$ is a natural bijection $\text{hom}_C(r, d) \rightarrow \text{hom}_D(c, Sd)$.*

Proof 1. If $\langle r, u \rangle$ is an universal arrow, then bijectivity of ψ_d is just a reformulation of the definition. It is also clear that ψ_d is natural in d , because if $g : d \rightarrow d'$ is a morphism, then $S(g' \circ f') \circ u = (Sg') \circ (Sg) \circ u$.

2. Now assume that $\psi_d : \text{hom}_C(r, d) \rightarrow \text{hom}_D(c, Sd)$ is a bijection for each d , and choose in particular $r = d$. Define $u := \psi_r(\text{id}_r)$, then $u : c \rightarrow Sr$ is a morphism in D . Consider this diagram for an arbitrary $f' : r \rightarrow d$

$$\begin{array}{ccc}
 \text{hom}_C(r, r) & \xrightarrow{\psi_r} & \text{hom}_D(c, Sr) \\
 \text{hom}_C(r, f') \downarrow & & \downarrow \text{hom}_D(s, Sf') \\
 \text{hom}_C(r, d) & \xrightarrow{\psi_d} & \text{hom}_D(c, Sd)
 \end{array}$$

Given a morphism $f : c \rightarrow Sd$ in D , there exists a unique morphism $f' : r \rightarrow d$ such that $f = \psi_d(f')$, because ψ_d is a bijection. Then we have

$$\begin{aligned}
 f &= \psi_d(f') \\
 &= (\psi_d \circ \text{hom}_C(r, f'))(\text{id}_r) \\
 &= (\text{hom}_D(c, Sf') \circ \psi_r)(\text{id}_r) && \text{(commutativity)} \\
 &= \text{hom}_D(c, Sf') \circ u && (u = \psi_r(\text{id}_r)) \\
 &= (Sf') \circ u.
 \end{aligned}$$

⊥

Universal arrows will be used now for a characterization of adjunctions in terms of natural transformations (we will sometimes omit the indices for the natural transformation φ that comes with an adjunction).

Theorem 1.109 *Let (F, G, φ) be an adjunction for the functors $F : K \rightarrow L$ and $G : L \rightarrow K$. Then there exist natural transformations $\eta : \text{Id}_K \rightarrow G \circ F$ and $\varepsilon : F \circ G \rightarrow \text{Id}_L$ with these properties:*

1. the pair $\langle \mathbf{F}x, \eta_x \rangle$ is a universal arrow from x to \mathbf{G} for each x in \mathbf{K} , and $\varphi(f) = \mathbf{G}f \circ \eta_x$ holds for each $f : \mathbf{F}x \rightarrow \mathbf{a}$,
2. the pair $\langle \mathbf{G}a, \varepsilon_a \rangle$ is universal from \mathbf{F} to \mathbf{a} for each a in \mathbf{L} , and $\varphi^{-1}(g) = \varepsilon_a \circ \mathbf{F}g$ holds for each $g : x \rightarrow \mathbf{G}a$,
3. the composites

$$\begin{array}{ccc}
 \mathbf{G} & \xrightarrow{\eta_{\mathbf{G}}} & \mathbf{G} \circ \mathbf{F} \circ \mathbf{G} \\
 & \searrow \text{id}_{\mathbf{G}} & \downarrow \mathbf{G}\varepsilon \\
 & & \mathbf{G}
 \end{array}
 \qquad
 \begin{array}{ccc}
 \mathbf{F} & \xrightarrow{\mathbf{F}\eta} & \mathbf{F} \circ \mathbf{G} \circ \mathbf{F} \\
 & \searrow \text{id}_{\mathbf{F}} & \downarrow \varepsilon_{\mathbf{F}} \\
 & & \mathbf{F}
 \end{array}$$

are the identities for \mathbf{G} resp. \mathbf{F} .

Proof 1. Put $\eta_x := \varphi_{x, \mathbf{F}x}(\text{id}_{\mathbf{F}x})$, then $\eta_x : x \rightarrow \mathbf{G}\mathbf{F}x$. In order to show that $\langle \mathbf{F}x, \eta_x \rangle$ is a universal arrow from x to \mathbf{G} , we take a morphism $f : x \rightarrow \mathbf{G}a$ for some object a in \mathbf{L} . Since $(\mathbf{F}, \mathbf{G}, \varphi)$ is an adjunction, we know that there exists a unique morphism $f' : \mathbf{F}x \rightarrow a$ such that $\varphi_{x,a}(f') = f$. We have also this commutative diagram

$$\begin{array}{ccc}
 \text{hom}_{\mathbf{K}}(\mathbf{F}x, \mathbf{F}x) & \xrightarrow{\varphi_{x, \mathbf{F}x}} & \text{hom}_{\mathbf{L}}(x, \mathbf{G}\mathbf{F}x) \\
 \text{hom}(\mathbf{F}x, f') \downarrow & & \downarrow \text{hom}_{\mathbf{L}}(x, \mathbf{G}f') \\
 \text{hom}_{\mathbf{K}}(\mathbf{F}x, a) & \xrightarrow{\varphi_{x,a}} & \text{hom}_{\mathbf{L}}(x, \mathbf{G}a)
 \end{array}$$

Thus

$$\begin{aligned}
 (\mathbf{G}f') \circ \eta_x &= (\text{hom}_{\mathbf{L}}(x, \mathbf{G}f') \circ \varphi_{x, \mathbf{F}x})(\text{id}_{\mathbf{F}x}) \\
 &= (\varphi_{x,a} \circ \text{hom}_{\mathbf{K}}(\mathbf{F}x, f'))(\text{id}_{\mathbf{F}x}) \\
 &= \varphi_{x,a}(f') \\
 &= f
 \end{aligned}$$

2. $\eta : \text{Id}_{\mathbf{K}} \rightarrow \mathbf{G} \circ \mathbf{F}$ is a natural transformation. Let $h : x \rightarrow y$ be a morphism in \mathbf{K} , then we have by Lemma 1.106

$$\begin{aligned}
 \mathbf{G}(\mathbf{F}h) \circ \eta_x &= \mathbf{G}(\mathbf{F}h) \circ \varphi_{x, \mathbf{F}x}(\text{id}_{\mathbf{F}x}) \\
 &= \varphi_{x, \mathbf{F}y}(\mathbf{F}h \circ \text{id}_{\mathbf{F}x}) \\
 &= \varphi_{x, \mathbf{F}y}(\text{id}_{\mathbf{F}y} \circ \mathbf{F}h) \\
 &= \varphi_{y, \mathbf{F}y}(\text{id}_{\mathbf{F}y}) \circ h \\
 &= \eta_y \circ h.
 \end{aligned}$$

3. Put $\varepsilon_a := \varphi_{\mathbf{G}a, a}^{-1}(\text{id}_{\mathbf{G}a})$ for the object a in \mathbf{L} , then the properties for ε are proved in exactly the same way as for those of η .

4. From $\varphi_{x,a}(f) = \mathbf{G}f \circ \eta_x$ we obtain

$$\text{id}_{\mathbf{G}a} = \varphi(\varepsilon_a) = \mathbf{G}\varepsilon_a \circ \eta_{\mathbf{G}a} = (\mathbf{G}\varepsilon \circ \eta_{\mathbf{G}})(a),$$

so that $\mathbf{G}\varepsilon \circ \eta\mathbf{G}$ is the identity transformation on \mathbf{G} . Similarly, $\eta\mathbf{F} \circ \mathbf{F}\varepsilon$ is the identity for \mathbf{F} .
 \dashv

The transformation η is sometimes called the *unit* of the adjunction, whereas ε is called its *counit*. The converse to Theorem 1.109 holds as well: from two transformations η and ε with the signatures as above one can construct an adjunction. The proof is a fairly straightforward verification.

Proposition 1.110 *Let $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$ and $\mathbf{G} : \mathbf{L} \rightarrow \mathbf{K}$ be functors, and assume that natural transformations $\eta : \text{Id}_{\mathbf{K}} \rightarrow \mathbf{G} \circ \mathbf{F}$ and $\varepsilon : \mathbf{F} \circ \mathbf{G} \rightarrow \text{Id}_{\mathbf{L}}$ are given so that $(\mathbf{G}\varepsilon) \circ (\eta\mathbf{G})$ is the identity of \mathbf{G} , and $(\varepsilon\mathbf{F}) \circ (\mathbf{F}\eta)$ is the identity of \mathbf{F} . Define $\varphi_{x,a}(k) := (\mathbf{G}k) \circ \eta_x$, whenever $k : \mathbf{F}x \rightarrow a$ is a morphism in \mathbf{L} . Then $(\mathbf{F}, \mathbf{G}, \varphi)$ defines an adjunction.*

Proof 1. Define $\theta_{x,a}(\ell) := \varepsilon_a \circ \mathbf{F}g$ for $\ell : x \rightarrow \mathbf{G}a$, then we have

$$\begin{aligned} \varphi_{x,a}(\theta_{x,a}(g)) &= \mathbf{G}(\varepsilon_a \circ \mathbf{F}g) \circ \eta_x \\ &= (\mathbf{G}\varepsilon_a) \circ (\mathbf{G}\mathbf{F}g) \circ \eta_x \\ &= (\mathbf{G}\varepsilon_a) \circ \eta_{\mathbf{G}a} \circ g && (\eta \text{ is natural}) \\ &= ((\mathbf{G}\varepsilon \circ \eta\mathbf{G})a) \circ g \\ &= \text{id}_{\mathbf{G}a}g \\ &= g \end{aligned}$$

Thus $\varphi_{x,a} \circ \theta_{x,a} = \text{id}_{\text{hom}_{\mathbf{L}}(x, \mathbf{G}a)}$. Similarly, one shows that $\theta_{x,a} \circ \varphi_{x,a} = \text{id}_{\text{hom}_{\mathbf{K}}(\mathbf{F}x, a)}$, so that $\varphi_{x,a}$ is a bijection.

2. We have to show that $\varphi_{x,a}$ is natural for each x, a , so take a morphism $f : a \rightarrow b$ in \mathbf{L} and chase $k : \mathbf{F}x \rightarrow a$ through this diagram.

$$\begin{array}{ccc} \text{hom}_{\mathbf{L}}(\mathbf{F}x, a) & \xrightarrow{\varphi_{x,a}} & \text{hom}_{\mathbf{K}}(x, \mathbf{G}a) \\ f_* \downarrow & & \downarrow (\mathbf{G}f)_* \\ \text{hom}_{\mathbf{L}}(\mathbf{F}x, b) & \xrightarrow{\varphi_{x,b}} & \text{hom}_{\mathbf{K}}(x, \mathbf{G}b) \end{array}$$

Then $((\mathbf{G}f)_* \circ \varphi_{x,a})(k) = (\mathbf{G}f \circ \mathbf{G}k) \circ \eta_x = \mathbf{G}(f \circ k) \circ \eta_x = \varphi_{x,b}(f_* \circ k)$.

\dashv

Thus for identifying an adjunction it is sufficient to identify its unit and its counit. This includes verifying the identity laws of the functors for the corresponding compositions. The following example has another look at currying (Example 1.103), demonstrating the approach and suggesting that identifying unit and counit is sometimes easier than working with the originally given definition.

Example 1.111 Continuing Example 1.103, we take the definitions of the endofunctors \mathbf{F} and \mathbf{G} from there. Define for the set X the natural transformations $\eta : \text{Id}_{\mathbf{Set}} \rightarrow \mathbf{G} \circ \mathbf{F}$ and $\varepsilon : \mathbf{F} \circ \mathbf{G} \rightarrow \text{Id}_{\mathbf{Set}}$ through

$$\eta_x : \begin{cases} X & \rightarrow (X \times E)^E \\ x & \mapsto \lambda e. \langle x, e \rangle \end{cases}$$

and

$$\varepsilon_X : \begin{cases} (X \times E)^E \times E & \rightarrow X \\ \langle g, e \rangle & \mapsto g(e) \end{cases}$$

Note that we have $(\mathbf{G}f)(h) = f \circ h$ for $f : X^E \rightarrow Y^E$ and $h \in X^E$, so that we obtain

$$\begin{aligned} \mathbf{G}\varepsilon_X(\eta_{\mathbf{G}X}(g))(e) &= (\varepsilon_X \circ \eta_{\mathbf{G}X}(g))(e) \\ &= \varepsilon_X(\eta_{\mathbf{G}X}(g))(e) \\ &= \varepsilon_X(\eta_{\mathbf{G}X}(g)(e)) \\ &= \varepsilon_X(g, e) \\ &= g(e), \end{aligned}$$

whenever $e \in E$ and $g \in \mathbf{G}X = X^E$, hence $(\mathbf{G}\varepsilon) \circ (\eta\mathbf{G}) = \text{id}_{\mathbf{G}}$. One shows similarly that $(\varepsilon\mathbf{F}) \circ (\mathbf{F}\eta) = \text{id}_{\mathbf{F}}$ through

$$\varepsilon_{\mathbf{F}X}(\mathbf{F}\eta_X(x, e)) = \eta_X(x)(e) = \langle x, e \rangle.$$

✎

Now let $(\mathbf{F}, \mathbf{G}, \varphi)$ be an adjunction with functors $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$ and $\mathbf{G} : \mathbf{L} \rightarrow \mathbf{K}$, the unit η and the counit ε . Define the functor \mathbf{T} through $\mathbf{T} := \mathbf{G} \circ \mathbf{F}$. Then $\mathbf{T} : \mathbf{K} \rightarrow \mathbf{K}$ defines an endofunctor on category \mathbf{K} with $\mu_a := (\mathbf{G}\varepsilon\mathbf{F})(a) = \mathbf{G}\varepsilon_{\mathbf{F}a}$ as a morphism $\mu_a : \mathbf{T}^2(a) \rightarrow \mathbf{T}a$. Because $\varepsilon_a : \mathbf{F}\mathbf{G}a \rightarrow a$ is a morphism in \mathbf{L} , and because $\varepsilon : \mathbf{F} \circ \mathbf{G} \rightarrow \text{Id}_{\mathbf{L}}$ is natural, the diagram

$$\begin{array}{ccc} (\mathbf{F} \circ \mathbf{G} \circ \mathbf{F} \circ \mathbf{G})a & \xrightarrow{\varepsilon_{(\mathbf{F} \circ \mathbf{G})a}} & (\mathbf{F} \circ \mathbf{G})a \\ (\mathbf{F} \circ \mathbf{G})\varepsilon_a \downarrow & & \downarrow \varepsilon_a \\ (\mathbf{F} \circ \mathbf{G})a & \xrightarrow{\varepsilon_a} & a \end{array}$$

is commutative. This means that this diagram

$$\begin{array}{ccc} \mathbf{F} \circ \mathbf{G} \circ \mathbf{F} \circ \mathbf{G} & \xrightarrow{\varepsilon_{(\mathbf{F} \circ \mathbf{G})}} & \mathbf{F} \circ \mathbf{G} \\ (\mathbf{F} \circ \mathbf{G})\varepsilon \downarrow & & \downarrow \varepsilon \\ \mathbf{F} \circ \mathbf{G} & \xrightarrow{\varepsilon} & \text{Id}_{\mathbf{K}} \end{array}$$

of functors and natural transformations commutes. Multiplying from the left with \mathbf{G} and from the right with \mathbf{F} gives this diagram.

$$\begin{array}{ccc} \mathbf{G} \circ \mathbf{F} \circ \mathbf{G} \circ \mathbf{F} \circ \mathbf{G} \circ \mathbf{F} & \xrightarrow{\mathbf{G}\varepsilon_{(\mathbf{F} \circ \mathbf{G} \circ \mathbf{F})}} & \mathbf{G} \circ \mathbf{F} \circ \mathbf{G} \circ \mathbf{F} \\ (\mathbf{G} \circ \mathbf{F} \circ \mathbf{G})\varepsilon\mathbf{F} \downarrow & & \downarrow \mathbf{G}\varepsilon\mathbf{F} \\ \mathbf{G} \circ \mathbf{F} \circ \mathbf{G} \circ \mathbf{F} & \xrightarrow{\mathbf{G}\varepsilon\mathbf{F}} & \mathbf{G} \circ \mathbf{F} \end{array}$$

Because $\mathbf{T}\mu = (\mathbf{G} \circ \mathbf{F} \circ \mathbf{G})\varepsilon\mathbf{F}$, and $\mathbf{G}\varepsilon_{(\mathbf{F} \circ \mathbf{G} \circ \mathbf{F})} = \mu\mathbf{T}$, this diagram can be written as

$$\begin{array}{ccc} \mathbf{T}^3 & \xrightarrow{\mathbf{T}\mu} & \mathbf{T}^2 \\ \mu\mathbf{T} \downarrow & & \downarrow \mu \\ \mathbf{T}^2 & \xrightarrow{\mu} & \mathbf{T} \end{array}$$

This gives the commutativity of the left hand diagram in Definition 1.94. Because $\mathbf{G}\varepsilon \circ \eta\mathbf{G}$ is the identity on \mathbf{G} , we obtain

$$\mathbf{G}\varepsilon_{\mathbf{F}a} \circ \eta_{\mathbf{G}\mathbf{F}a} = (\mathbf{G}\varepsilon \circ \eta\mathbf{G})(\mathbf{F}a) = \mathbf{G}\mathbf{F}a,$$

which implies that the diagram

$$\begin{array}{ccc} \mathbf{T}a & \xrightarrow{\mu_{\mathbf{T}a}} & \mathbf{T}^2a \\ & \searrow \text{id}_{\mathbf{T}a} & \downarrow \mu_a \\ & & \mathbf{T}a \end{array}$$

commutes. On the other hand, we know that $\varepsilon\mathbf{F} \circ \mathbf{F}\eta$ is the identity on \mathbf{F} ; this yields

$$\mathbf{G}\varepsilon_{\mathbf{F}a} \circ \mathbf{G}\mathbf{F}\eta_a = \mathbf{G}(\varepsilon\mathbf{F} \circ \mathbf{F}\eta)a = \mathbf{G}\mathbf{F}a.$$

Hence we may complement the last diagram:

$$\begin{array}{ccccc} \mathbf{T}a & \xrightarrow{\mu_{\mathbf{T}a}} & \mathbf{T}^2a & \xleftarrow{\mathbf{T}\eta_a} & \mathbf{T}a \\ & \searrow \text{id}_{\mathbf{T}a} & \downarrow \mu_a & & \swarrow \text{id}_{\mathbf{T}a} \\ & & \mathbf{T}a & & \end{array}$$

This gives the right hand side diagram in Definition 1.94. We have shown

Proposition 1.112 *Each adjunction defines a monad.* \dashv

It turns out that we not only may proceed from an adjunction to a monad, but that it is also possible to traverse this path in the other direction. We will show that a monad defines an adjunction. In order to do that, we have to represent the functorial part of a monad as the composition of two other functors, so we need a second category for this. The algebras which are defined for a monad provide us with this category. So we will define algebras (and in a later chapter, their counterparts, coalgebras), and we will study them. This will help us in showing that each monad defines an adjunction. Finally, we will have a look at two examples for algebras, in order to illuminate this concept.

Given a monad (\mathbf{T}, η, μ) in a category \mathbf{K} , a pair $\langle x, h \rangle$ consisting of an object x and a morphism $h : \mathbf{T}x \rightarrow x$ in \mathbf{K} is called an *Eilenberg-Moore algebra* for the monad iff the following diagrams commute

$$\begin{array}{ccc} \mathbf{T}^2x & \xrightarrow{\mathbf{T}h} & \mathbf{T}x \\ \mu_x \downarrow & & \downarrow h \\ \mathbf{T}x & \xrightarrow{h} & x \end{array} \qquad \begin{array}{ccc} x & \xrightarrow{\eta_x} & \mathbf{T}x \\ & \searrow \text{id}_x & \downarrow h \\ & & x \end{array}$$

The morphism h is called the *structure morphism* of the algebra, x its carrier.

An *algebra morphism* $f : \langle x, h \rangle \rightarrow \langle x', h' \rangle$ between the algebras $\langle x, h \rangle$ and $\langle x', h' \rangle$ is a morphism $f : x \rightarrow x'$ in \mathbf{K} which renders the diagram

$$\begin{array}{ccc} \mathbf{T}x & \xrightarrow{h} & x \\ \mathbf{T}f \downarrow & & \downarrow f \\ \mathbf{T}x' & \xrightarrow{h'} & x' \end{array}$$

commutative. Eilenberg-Moore algebras together with their morphisms form a category $\mathbf{Alg}_{(\mathbf{T}, \eta, \mu)}$. We will usually omit the reference to the monad. Fix for the moment (\mathbf{T}, η, μ) as a monad in category \mathbf{K} , and let $\mathbf{Alg} := \mathbf{Alg}_{(\mathbf{T}, \eta, \mu)}$ be the associated category of Eilenberg-Moore algebras.

We give some simple examples.

Lemma 1.113 *The pair $\langle \mathbf{T}x, \mu_x \rangle$ is a \mathbf{T} -algebra for each x in \mathbf{K} .*

Proof This is immediate from the laws for η and μ in a monad. \dashv

These algebras are usually called the *free algebras* for the monad. Morphisms in the base category \mathbf{K} translate into morphisms in \mathbf{Alg} through functor \mathbf{T} .

Lemma 1.114 *If $f : x \rightarrow y$ is a morphism in \mathbf{K} , then $\mathbf{T}f : \langle \mathbf{T}x, \mu_x \rangle \rightarrow \langle \mathbf{T}y, \mu_y \rangle$ is a morphism in \mathbf{Alg} . If $\langle x, h \rangle$ is an algebra, then $h : \langle \mathbf{T}x, \mu_x \rangle \rightarrow \langle x, h \rangle$ is a morphism in \mathbf{Alg} .*

Proof Because $\mu : \mathbf{T}^2 \rightarrow \mathbf{T}$ is a natural transformation, we see $\mu_y \circ \mathbf{T}^2 f = (\mathbf{T}f) \circ \mu_x$. This is just the defining equation for a morphism in \mathbf{Alg} . The second assertion follows also from the defining equation of an algebra morphism. \dashv

We will identify the algebras for the power set monad now, which are closely connected to semi-lattices. Recall that an ordered set (X, \leq) is a *sup-semi lattice* iff each subset has its supremum in X .

Example 1.115 The algebras for the monad (\mathcal{P}, η, μ) in the category \mathbf{Set} of sets with maps (sometimes called the *Manes monad*) may be identified with the complete sup-semi lattices. We will show this now.

Assume first that \leq is a partial order on a set X that is sup-complete, so that $\sup A$ exists for each $A \subseteq X$. Define $h(A) := \sup A$, then we have for each $\mathcal{A} \in \mathcal{P}(\mathcal{P}(X))$ from the familiar properties of the supremum

$$\sup(\bigcup \mathcal{A}) = \sup \{\sup a \mid a \in \mathcal{A}\}.$$

This translates into $(h \circ \mu_X)(\mathcal{A}) = (h \circ (\mathcal{P}h))(\mathcal{A})$. Because $x = \sup\{x\}$ holds for each $x \in X$, we see that $\langle X, h \rangle$ defines an algebra.

Assume on the other hand that $\langle X, h \rangle$ is an algebra, and put

$$x \leq x' \Leftrightarrow h(\{x, x'\}) = x'$$

for $x, x' \in X$. This defines a partial order: reflexivity and antisymmetry are obvious. Transitivity is seen as follows: assume $x \leq x'$ and $x' \leq x''$, then

$$\begin{aligned} h(\{x, x''\}) &= h(\{h(\{x\}), h(\{x', x''\})\}) = (h \circ (\mathcal{P}h))(\{\{x\}, \{x', x''\}\}) \\ &= (h \circ \mu_X)(\{\{x\}, \{x', x''\}\}) = h(\{x, x', x''\}) = (h \circ \mu_X)(\{\{x, x'\}, \{x', x''\}\}) \\ &= (h \circ (\mathcal{P}h))(\{\{x, x'\}, \{x', x''\}\}) = h(\{x', x''\}) = x''. \end{aligned}$$

It is clear from $\{x\} \cup \emptyset = \{x\}$ for every $x \in X$ that $h(\emptyset)$ is the smallest element. Finally, it has to be shown that $h(A)$ is the smallest upper bound for $A \subseteq X$ in the order \leq . We may

assume that $A \neq \emptyset$. Suppose that $x \leq t$ holds for all $x \in A$, then

$$\begin{aligned} h(A \cup \{t\}) &= h\left(\bigcup_{x \in A} \{x, t\}\right) = (h \circ \mu_x)(\{\{x, t\} \mid x \in A\}) \\ &= (h \circ (\mathcal{P}h))(\{\{x, t\} \mid x \in A\}) = h(\{h(\{x, t\}) \mid x \in A\}) = h(\{t\}) = t. \end{aligned}$$

Thus, if $x \leq t$ for all $x \in A$, hence $h(A) \leq t$, thus $h(A)$ is an upper bound to A , and similarly, $h(A)$ is the smallest upper bound. \mathbb{M}

We have shown that each adjunction defines a monad, and now turn to the converse. In fact, we will show that each monad defines an adjunction the monad of which is the given monad. Fix the monad (\mathbf{T}, η, μ) over category \mathbf{K} , and define as above $\mathbf{Alg} := \mathbf{Alg}_{(\mathbf{T}, \eta, \mu)}$ as the category of Eilenberg-Moore algebras. We want to define an adjunction, so by Proposition 1.110 it will be most convenient approach to solve the problem by defining unit and counit, after the corresponding functors have been identified.

Lemma 1.116 *Define $\mathbf{F}a := \langle \mathbf{T}a, \mu_a \rangle$ for the object $a \in |\mathbf{K}|$, and if $f : a \rightarrow b$ is a morphism in \mathbf{K} , define $\mathbf{F}f := \mathbf{T}f$. Then $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{Alg}$ is a functor.*

Proof We have to show that $\mathbf{F}f : \langle \mathbf{T}a, \mu_a \rangle \rightarrow \langle \mathbf{T}b, \mu_b \rangle$ is an algebra morphism. Since $\mu : \mathbf{T}^2 \rightarrow \mathbf{T}$ is natural, we obtain this commutative diagram

$$\begin{array}{ccc} \mathbf{T}a & \xrightarrow{\mu_a} & \mathbf{T}^2a \\ \mathbf{T}f \downarrow & & \downarrow \mathbf{T}^2f \\ \mathbf{T}b & \xrightarrow{\mu_b} & \mathbf{T}^2b \end{array}$$

But this is just the defining definition for an algebra morphism. \dashv

This statement is trivial:

Lemma 1.117 *Given an Eilenberg-Moore algebra $\langle x, h \rangle \in |\mathbf{Alg}|$, define $\mathbf{G}(x, h) := x$; if $f : \langle x, h \rangle \rightarrow \langle x', h' \rangle$ is a morphism in \mathbf{Alg} , put $\mathbf{G}f := f$. Then $\mathbf{G} : \mathbf{Alg} \rightarrow \mathbf{K}$ is a functor. Moreover we have $\mathbf{G} \circ \mathbf{F} = \mathbf{T}$. \dashv*

We require two natural transformations, which are defined now, and which are intended to serve as the unit and as the counit, respectively, for the adjunction. We define for the unit η the originally given η , so that $\eta : \text{Id}_{\mathbf{K}} \rightarrow \mathbf{G} \circ \mathbf{F}$ is a natural transformation. The counit ε is defined through $\varepsilon_{\langle x, h \rangle} := h$, so that $\varepsilon_{\langle x, h \rangle} : (\mathbf{F} \circ \mathbf{G})(x, h) \rightarrow \text{Id}_{\mathbf{Alg}}(x, h)$. This defines a natural transformation $\varepsilon : \mathbf{F} \circ \mathbf{G} \rightarrow \text{Id}_{\mathbf{Alg}}$. In fact, let $f : \langle x, h \rangle \rightarrow \langle x', h' \rangle$ be a morphism in \mathbf{Alg} , then — by expanding definitions — the diagram on the left hand side translates to the one on the right hand side, which commutes:

$$\begin{array}{ccc} (\mathbf{F} \circ \mathbf{G})(x, h) & \xrightarrow{\varepsilon_{\langle x, h \rangle}} & \langle x, h \rangle \\ (\mathbf{F} \circ \mathbf{G})f \downarrow & & \downarrow f \\ (\mathbf{F} \circ \mathbf{G})(x', h') & \xrightarrow{\varepsilon_{\langle x', h' \rangle}} & \langle x', h' \rangle \end{array} \qquad \begin{array}{ccc} \langle \mathbf{T}x, \mu_x \rangle & \xrightarrow{h} & \langle x, h \rangle \\ \mathbf{T}f \downarrow & & \downarrow f \\ \langle \mathbf{T}x', \mu_{x'} \rangle & \xrightarrow{h'} & \langle x', h' \rangle \end{array}$$

Now take an object $a \in |\mathbf{K}|$, then

$$(\varepsilon \mathbf{F} \circ \mathbf{F} \eta)(a) = \varepsilon_{\mathbf{F}a}(\mathbf{F} \eta_a) = \varepsilon_{\langle \mathbf{T}a, \mu_a \rangle}(\mathbf{T} \eta_a) = \mu_a(\mathbf{T} \eta_a) = \text{id}_{\mathbf{F}a}.$$

On the other hand, we have for the algebra $\langle x, h \rangle$

$$(\mathbf{G}\varepsilon \circ \eta\mathbf{G})(x, h) = \mathbf{G}\varepsilon_{\langle x, h \rangle}(\eta_{\mathbf{G}\langle x, h \rangle}) = \mathbf{G}\varepsilon_{\langle x, h \rangle}(\eta_x) = \varepsilon_{\langle x, h \rangle}(\eta_x) = h\eta_x \stackrel{(*)}{=} \text{id}_x = \text{id}_{\mathbf{G}\langle x, h \rangle}$$

where $(*)$ uses that $h : \mathbf{T}x \rightarrow x$ is the structure morphism of an algebra. Taken together, we see that η and ε satisfy the requirements of unit and counit for an adjunction according to Proposition 1.110.

Hence we have nearly established

Proposition 1.118 *Every monad defines an adjunction. The monad defined by the adjunction is the original one.*

Proof We have only to prove the last assertion. But this is trivial, because $(\mathbf{G}\varepsilon\mathbf{F})a = (\mathbf{G}\varepsilon)(\mathbf{T}a, \mu_a) = \mathbf{G}\mu_a = \mu_a$. \dashv

Algebras for discrete probabilities. We identify now the algebras for the functor \mathbf{D} which assigns to each set its discrete subprobabilities with finite support, see Example 1.69. Some preliminary and motivating observations are made first.

Put

$$\Omega := \{\langle \alpha_1, \dots, \alpha_k \rangle \mid k \in \mathbb{N}, \alpha_i \geq 0, \sum_{i=1}^k \alpha_i \leq 1\}$$

as the set of all positive convex coefficients, and call a subset V of a real vector space *positive convex* iff $\sum_{i=1}^k \alpha_i \cdot x_i \in X$ for $x_1, \dots, x_k \in X$, $\langle \alpha_1, \dots, \alpha_k \rangle \in \Omega$. Positive convexity appears to be related to subprobabilities: if $\sum_{i=1}^k \alpha_i \cdot x_i$ is perceived as an observation in which item x_i is assigned probability α_i , then clearly $\sum_{i=1}^k \alpha_i \leq 1$ under the assumption that the observation is incomplete, i.e., that not every possible case has been realized.

Suppose a set X over which we formulate subprobabilities is embedded as a positive convex set into a linear space V over the reals as a positive convex structure. In this case we could read off a positive convex combination for an element the probabilities with which the respective components occurs.

These observations meet the intuition about positive convexity, but it has the drawback that we have to look for a linear space V into which X to embed. It has the additional shortcoming that once we did identify V , the positive convex structure on X is fixed through the vector space, but we will see soon that we need some flexibility. Consequently, we propose an abstract description of positive convexity, much in the spirit of Pumplün's approach [Pum03]. Thus the essential properties (for us, that is) of positive convexity are described intrinsically for X without having to resort to a vector space. This leads to the definition of a positive convex structure.

Definition 1.119 *A positive convex structure \wp on a set X has for each $\alpha = \langle \alpha_1, \dots, \alpha_n \rangle \in \Omega$ a map $\alpha_\wp : X^n \rightarrow X$ which we write as*

$$\alpha_\wp(x_1, \dots, x_n) = \sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot x_i,$$

such that

★ $\sum_{1 \leq i \leq n}^{\wp} \delta_{i,k} \cdot x_i = x_k$, where $\delta_{i,j}$ is Kronecker's δ (thus $\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$, otherwise),

☆ the identity

$$\sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot \left(\sum_{1 \leq k \leq m} \beta_{i,k} \cdot x_k \right) = \sum_{1 \leq k \leq m}^{\wp} \left(\sum_{1 \leq i \leq n} \alpha_i \beta_{i,k} \right) \cdot x_k$$

holds whenever $\langle \alpha_1, \dots, \alpha_n \rangle, \langle \beta_{i,1}, \dots, \beta_{i,m} \rangle \in \Omega, 1 \leq i \leq n$.

Property ★ looks quite trivial, when written down this way. Rephrasing, it states that the map

$$\langle \delta_{1,k}, \dots, \delta_{n,k} \rangle_{\wp} : T^n \rightarrow T,$$

which is assigned to the n -tuple $\langle \delta_{1,k}, \dots, \delta_{n,k} \rangle$ through \wp acts as the projection to the k^{th} component for $1 \leq k \leq n$. Similarly, property ☆ may be re-coded in a formal but less concise way. Thus we will use freely the notation from vector spaces, omitting in particular the explicit reference to the structure whenever possible. Hence simple addition $\alpha_1 \cdot x_1 + \alpha_2 \cdot x_2$ will be written rather than $\sum_{1 \leq i \leq 2}^{\wp} \alpha_i \cdot x_i$, with the understanding that it refers to a given positive convex structure \wp on X .

It is an easy exercise to establish that for a positive convex structure the usual rules for manipulating sums in vector spaces apply, e.g., $1 \cdot x = x$, $\sum_{i=1}^n \alpha_i \cdot x_i = \sum_{i=1, \alpha_i \neq 0}^n \alpha_i \cdot x_i$, or the law of associativity, $(\alpha_1 \cdot x_1 + \alpha_2 \cdot x_2) + \alpha_3 \cdot x_3 = \alpha_1 \cdot x_1 + (\alpha_2 \cdot x_2 + \alpha_3 \cdot x_3)$. Nevertheless, care should be observed, for of course not all rules apply: we cannot in general conclude $x = x'$ from $\alpha \cdot x = \alpha \cdot x'$, even if $\alpha \neq 0$.

A morphism $\theta : \langle X_1, \wp_1 \rangle \rightarrow \langle X_2, \wp_2 \rangle$ between positive convex structures is a map $\theta : X_1 \rightarrow X_2$ such that

$$\theta \left(\sum_{1 \leq i \leq n}^{\wp_1} \alpha_i \cdot x_i \right) = \sum_{1 \leq i \leq n}^{\wp_2} \alpha_i \cdot \theta(x_i)$$

holds for $x_1, \dots, x_n \in X$ and $\langle \alpha_1, \dots, \alpha_n \rangle \in \Omega$. In analogy to linear algebra, θ will be called an *affine* map. Positive convex structures with their morphisms form a category **StrConv**.

We need some technical preparations, which are collected in the following

Lemma 1.120 *Let X and Y be sets.*

1. *Given a map $f : X \rightarrow Y$, let $p = \alpha_1 \cdot \delta_{a_1} + \dots + \alpha_n \cdot \delta_{a_n}$ be the linear combination of Dirac measures for $x_1, \dots, x_n \in X$ with positive convex $\langle \alpha_1, \dots, \alpha_n \rangle \in \Omega$. Then $\mathbf{D}(f)(p) = \alpha_1 \cdot \delta_{f(x_1)} + \dots + \alpha_n \cdot \delta_{f(x_n)}$.*
2. *Let p_1, \dots, p_n be discrete subprobabilities X , and let $M = \alpha_1 \cdot \delta_{p_1} + \dots + \alpha_n \cdot \delta_{p_n}$ be the linear combination of the corresponding Dirac measures in $(\mathbf{D} \circ \mathbf{D})X$ with positive convex coefficients $\langle \alpha_1, \dots, \alpha_n \rangle \in \Omega$. Then $\mu_X(M) = \alpha_1 \cdot p_1 + \dots + \alpha_n \cdot p_n$.*

Proof The first part follows directly from the observation $\mathbf{D}(f)(\delta_x)(B) = \delta_x(f^{-1}[B]) = \delta_{f(x)}(B)$, and the second one is easily inferred from the formula for μ in Example 1.98. \dashv

The algebras are described now without having to resort to $\mathbf{S}(X)$ through an intrinsic characterization using positive convex structures with affine maps. This characterization is compa-

rable to the one given by Manes for the power set monad (which also does not resort explicitly to the underlying monad or its functor); see Example 1.115.

Lemma 1.121 *Given an algebra $\langle X, h \rangle$, define for $x_1, \dots, x_n \in X$ and the positive convex coefficients $\langle \alpha_1, \dots, \alpha_n \rangle \in \Omega$*

$$\langle \alpha_1, \dots, \alpha_n \rangle_{\wp} := \sum_{i=1}^n \alpha_i \cdot x_i := h\left(\sum_{i=1}^n \alpha_i \cdot \delta_{x_i}\right).$$

This defines a positive convex structure \wp on X .

Proof 1. Because

$$h\left(\sum_{i=1}^n \delta_{i,j} \cdot \delta_{x_i}\right) = h(\delta_{x_j}) = x_j,$$

property \star in Definition 1.119 is satisfied.

2. Proving property \star , we resort to the properties of an algebra and a monad:

$$\sum_{i=1}^n \alpha_i \cdot \left(\sum_{k=1}^m \beta_{i,k} \cdot x_k\right) = h\left(\sum_{i=1}^n \alpha_i \cdot \delta_{\sum_{k=1}^m \beta_{i,k} \cdot x_k}\right) \quad (2)$$

$$= h\left(\sum_{i=1}^n \alpha_i \cdot \delta_{h\left(\sum_{k=1}^m \beta_{i,k} \cdot \delta_{x_k}\right)}\right) \quad (3)$$

$$= h\left(\sum_{i=1}^n \alpha_i \cdot \mathbf{S}(h)\left(\delta_{\sum_{k=1}^m \beta_{i,k} \cdot \delta_{x_k}}\right)\right) \quad (4)$$

$$= (h \circ \mathbf{S}(h))\left(\sum_{i=1}^n \alpha_i \cdot \delta_{\sum_{k=1}^m \beta_{i,k} \cdot \delta_{x_k}}\right) \quad (5)$$

$$= (h \circ \mu_X)\left(\sum_{i=1}^n \alpha_i \cdot \delta_{\sum_{k=1}^m \beta_{i,k} \cdot \delta_{x_k}}\right) \quad (6)$$

$$= h\left(\sum_{i=1}^n \alpha_i \cdot \mu_X\left(\delta_{\sum_{k=1}^m \beta_{i,k} \cdot \delta_{x_k}}\right)\right) \quad (7)$$

$$= h\left(\sum_{i=1}^n \alpha_i \cdot \left(\sum_{k=1}^m \beta_{i,k} \cdot \delta_{x_k}\right)\right) \quad (8)$$

$$= h\left(\sum_{k=1}^m \left(\sum_{i=1}^n \alpha_i \cdot \beta_{i,k}\right) \delta_{x_k}\right) \quad (9)$$

$$= \sum_{k=1}^m \left(\sum_{i=1}^n \alpha_i \cdot \beta_{i,k}\right) x_k. \quad (10)$$

The equations (2) and (3) reflect the definition of the structure, equation (4) applies $\delta_{h(\tau)} = \mathbf{S}(h)(\delta_\tau)$, equation (5) uses the linearity of $\mathbf{S}(h)$ according to Lemma 1.120, equation (6) is due to h being an algebra. Winding down, equation (7) uses Lemma 1.120 again, this time for μ_X , equation (8) uses that $\mu_X \circ \delta_\tau = \tau$, equation (9) is just rearranging terms, and equation (10) is the definition again. \dashv

The converse holds as well. Assume that we have a positive convex structure \wp on X . Put

$$h\left(\sum_{i=1}^n \alpha_i \cdot \delta_{x_i}\right) := \sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot x_i$$

for $\langle \alpha_1, \dots, \alpha_n \rangle \in \Omega$ and $x_1, \dots, x_n \in X$. One first checks that h is well-defined: This is so since

$$\sum_{i=1}^n \alpha_i \cdot \delta_{x_i} = \sum_{j=1}^m \alpha'_j \cdot \delta_{x'_j}$$

implies that

$$\sum_{i=1, \alpha_i \neq 0}^n \alpha_i \cdot \delta_{x_i} = \sum_{j=1, \alpha'_j \neq 0}^m \alpha'_j \cdot \delta_{x'_j},$$

hence given i with $\alpha_i \neq 0$ there exists j with $\alpha'_j \neq 0$ such that $x_i = x'_j$ with $\alpha_i = \alpha'_j$ and vice versa. Consequently,

$$\sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot x_i = \sum_{1 \leq i \leq n, \alpha_i \neq 0}^{\wp} \alpha_i \cdot x_i = \sum_{1 \leq j \leq n, \alpha'_j \neq 0}^{\wp} \alpha'_j \cdot x'_j = \sum_{1 \leq j \leq n}^{\wp} \alpha'_j \cdot x'_j$$

is inferred from the properties of positive convex structures. Thus $h : \mathbf{DX} \rightarrow X$.

An easy induction using property \star shows that h is an affine map, i.e., that we have

$$h\left(\sum_{i=1}^n \alpha_i \cdot \tau_i\right) = \sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot h(\tau_i) \quad (11)$$

for $\langle \alpha_1, \dots, \alpha_n \rangle \in \Omega$ and $\tau_1, \dots, \tau_n \in \mathbf{DX}$.

Now let $f = \sum_{i=1}^n \alpha_i \cdot \delta_{\tau_i} \in \mathbf{D}^2X$ with $\tau_1, \dots, \tau_n \in \mathbf{DX}$. Then we obtain from Lemma 1.120 that $\mu_X f = \sum_{i=1}^n \alpha_i \cdot \tau_i$. Consequently, we obtain from 11 that $h(\mu_X f) = \sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot h(\tau_i)$. On the other hand, Lemma 1.120 implies together with 11

$$\begin{aligned} (h \circ \mathbf{D}h)f &= h\left(\sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot (\mathbf{D}h)(\tau_i)\right) \\ &= \sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot h((\mathbf{D}h)(\tau_i)) \\ &= \sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot h(\delta_{h(\tau_i)}) \\ &= \sum_{1 \leq i \leq n}^{\wp} \alpha_i \cdot h(\tau_i), \end{aligned}$$

because $h(\delta_{h(\tau_i)}) = h(\tau_i)$. We infer from \star that $h \circ \mu_X = \text{id}_X$. Hence we have established

Proposition 1.122 *Each positive convex structure on X induces an algebra for \mathbf{DX} . \dashv*

Thus we obtain a complete characterization of the Eilenberg-Moore algebras for this monad.

Theorem 1.123 *The Eilenberg-Moore algebras for the discrete probability monad are exactly the positive convex structures. \dashv*

This characterization carries over to the probabilistic version of the monad; we leave the simple formulation to the reader. A similar characterization is possible for the continuous version of this functor, at least in Polish spaces. This requires a continuity condition, however.

1.6 Coalgebras

A coalgebra for a functor \mathbf{F} is characterized by a carrier object \mathbf{c} and by a morphism $\mathbf{c} \rightarrow \mathbf{F}\mathbf{c}$. This fairly general structure can be found in many applications, as we will see. So we will first define formally what a coalgebra is, and then provide a gallery of examples, some of them already discussed in another disguise, some of them new. The common thread is their formulation as a coalgebra. Bisimilar coalgebras will be discussed, indicating some interesting facets of the possibilities to describe behavioral equivalence of some sorts.

Definition 1.124 *Given the endofunctor \mathbf{F} on category \mathbf{K} , an object \mathbf{a} on \mathbf{K} together with a morphism $f : \mathbf{a} \rightarrow \mathbf{F}\mathbf{a}$ is called a coalgebra for \mathbf{K} . Morphism f is sometimes called the dynamics of the coalgebra, \mathbf{a} its carrier.*

Comparing the definitions of an algebra and a coalgebra, we see that for a coalgebra the functor \mathbf{F} is a arbitrary endofunctor on \mathbf{K} , which an algebra requires a monad and compatibility with unit and multiplication. This coalgebras are conceptually simpler by demanding less resources.

We are going to enter now the gallery of examples and start with coalgebras for the power set functor. This example will be with us for quite some time, in particular when we will interpret modal logics. A refinement of this example will be provided by labelled transition systems.

Example 1.125 We consider the category **Set** of sets with maps as morphisms and the power set functor \mathcal{P} . An \mathcal{P} coalgebra consists of a set A and a map $f : A \rightarrow \mathcal{P}(A)$. Hence we have $f(\mathbf{a}) \subseteq A$ for all $\mathbf{a} \in A$, so that a **Set**-coalgebra can be represented as a relation $\{\langle \mathbf{a}, \mathbf{b} \rangle \mid \mathbf{b} \in f(\mathbf{a}), \mathbf{a} \in A\}$ over A . If, conversely, $R \subseteq A \times A$ is a relation, then $f(\mathbf{a}) := \{\mathbf{b} \in A \mid \langle \mathbf{a}, \mathbf{b} \rangle \in R\}$ is a map $f : A \rightarrow \mathcal{P}(A)$. ✎

A slight extension is to be observed when we introduce actions, formally, labels for our transitions. Here a transition is dependent on an action which serves as a label to the corresponding relation.

Example 1.126 Let us interpret a labeled transition system $(S, (\rightsquigarrow_{\mathbf{a}})_{\mathbf{a} \in A})$ over state space S with set A of actions, see Example 1.68. Then $\rightsquigarrow_{\mathbf{a}} \subseteq S \times S$ for all actions $\mathbf{a} \in A$.

Working again in **Set**, we define for the set S and for the map $f : S \rightarrow \mathbf{T}$

$$\begin{aligned} \mathbf{T}S &:= \mathcal{P}(A \times S), \\ (\mathbf{T}f)(B) &:= \{\langle \mathbf{a}, f(\mathbf{x}) \rangle \mid \langle \mathbf{a}, \mathbf{x} \rangle \in B\} \end{aligned}$$

(hence $\mathbf{T} = \mathcal{P}(A \times -)$). Define $f(s) := \{\langle \mathbf{a}, s' \rangle \mid s \rightsquigarrow_{\mathbf{a}} s'\}$, thus $f : S \rightarrow \mathbf{T}S$ is a morphism in **Set**. Consequently, a transition system is interpreted as a coalgebra for the functor $\mathcal{P}(A \times -)$.

✎

Example 1.127 Let A be the inputs, B the outputs and X the states of an automaton with output, see Example 1.67. Put $\mathbf{F} := (- \times B)^A$. For $f : X \rightarrow Y$ we have this commutative diagram.

$$\begin{array}{ccc} & A & \\ t \swarrow & & \searrow (\mathbf{F}f)t \\ X \times B & \xrightarrow{f \times \text{id}_B} & Y \times B \end{array}$$

Let (S, f) be an \mathbf{F} -coalgebra, thus $f : S \rightarrow \mathbf{F}S = (S \times B)^A$. Input $a \in A$ in state $s \in S$ yields $f(s)(a) = \langle s', b \rangle$, so that s' is the new state, and b is the output. Hence automata with output are perceived as coalgebras, in this case for the functor $(- \times B)^A$. \upharpoonright

While the automata in Example 1.127 are deterministic (and completely specified), we can also use a similar approach to modelling nondeterministic automata.

Example 1.128 Let A, B, X be as in Example 1.127, but take this time $\mathbf{F} := \mathcal{P}(- \times B)^A$ as a functor, so that this diagram commutes for $f : X \rightarrow Y$:

$$\begin{array}{ccc} & A & \\ t \swarrow & & \searrow (\mathbf{F}f)t \\ \mathcal{P}(X \times B) & \xrightarrow{\mathcal{P}(f \times \text{id}_B)} & \mathcal{P}(Y \times B) \end{array}$$

Thus $\mathcal{P}(f \times B)(D) = \{\langle f(x), b \rangle \in Y \times B \mid \langle x, b \rangle \in D\}$. Then (S, g) is an \mathbf{F} coalgebra iff input $a \in A$ in state $s \in S$ gives $g(s)(a) \in S \times B$ as the set of possible new states and outputs.

As a variant, we can replace $\mathcal{P}(- \times B)$ by $\mathcal{P}_f(- \times B)$, so that the automaton presents only a finite number of alternatives. \upharpoonright

Binary trees may be modelled through coalgebras as well:

Example 1.129 Put $\mathbf{F}X := \{*\} + X \times X$, where $*$ is a new symbol. If $f : X \rightarrow Y$, put

$$\mathbf{F}(f)(t) := \begin{cases} *, & \text{if } t = * \\ \langle x_1, x_2 \rangle, & \text{if } t = \langle x_1, x_2 \rangle. \end{cases}$$

Then \mathbf{F} is an endofunctor on **Set**. Let (S, f) be an \mathbf{F} -coalgebra, then $f(s) \in \{*\} + S \times S$. This is interpreted that s is a leaf iff $f(s) = *$, and an inner node with offsprings $\langle s_1, s_2 \rangle$, if $f(s) = \langle s_1, s_2 \rangle$. Thus such a coalgebra represents a binary tree (which may be of infinite depth). \upharpoonright

The following example shows that probabilistic transitions may be modelled as coalgebras as well.

Example 1.130 Working in the category **Meas** of measurable spaces with measurable maps, we have introduced in Example 1.99 the subprobability functor \mathbb{S} as an endofunctor on **Meas**. Let (X, K) be a coalgebra for \mathbb{S} (we omit here the σ -algebra from the notation), then $K : X \rightarrow \mathbb{S}X$ is measurable, so that

1. $K(x)$ is a subprobability on (the measurable sets of) X ,
2. for each measurable set $D \subseteq X$, the map $x \mapsto K(x)(D)$ is measurable,

see Example 1.14 and Exercise 7. Thus K is a subprobabilistic transition kernel on X . \mathbb{M}

Let us have a look at the upper closed sets introduced in Example 1.71. Coalgebras for this functor will be used for an interpretation of games, see Example 1.190.

Example 1.131 Let $\mathbf{VS} := \{V \subseteq \mathcal{P}S \mid V \text{ is upper closed}\}$. This functor has been studied in Example 1.71. A coalgebra (S, f) for \mathbf{V} is a map $f : S \rightarrow \mathbf{VS}$, so that $f(s) \subseteq \mathcal{P}(S)$ is upper closed, hence $A \in f(s)$ and $B \supseteq A$ imply $b \in f(s)$ for each $s \in S$. We interpret $f(s)$ as the set of states a player may reach in state s , so that if the player can reach A and $A \subseteq B$, then the player certainly can reach B .

\mathbf{V} is the basis for neighborhood models in modal logics, see, e.g., [Che89, Ven07] and page 92. \mathbb{M}

It is natural to ask for morphisms of coalgebras, so that coalgebras can be related to each other. This is a fairly straightforward definition.

Definition 1.132 Let \mathbf{F} be an endofunctor on category \mathbf{K} , then $t : (a, f) \rightarrow (b, g)$ is a coalgebra morphism for the \mathbf{F} -coalgebras (a, f) and (b, g) iff $t : a \rightarrow b$ is a morphism in \mathbf{K} such that $g \circ t = \mathbf{F}(t) \circ f$.

Thus $t : (a, f) \rightarrow (b, g)$ is a coalgebra morphism iff $t : a \rightarrow b$ is a morphism so that this diagram commutes:

$$\begin{array}{ccc} a & \xrightarrow{t} & b \\ f \downarrow & & \downarrow g \\ \mathbf{F}a & \xrightarrow{\mathbf{F}t} & \mathbf{F}b \end{array}$$

It is clear that \mathbf{F} -coalgebras form a category with coalgebra morphisms as morphisms. We reconsider some previously discussed examples and shed some light on the morphisms for these coalgebras.

Example 1.133 Continuing Example 1.129 on binary trees, let $r : (S, f) \rightarrow (T, g)$ be a morphism for the \mathbf{F} -coalgebras (S, f) and (T, g) . Thus $g \circ r = \mathbf{F}(r) \circ f$. This entails

1. $f(s) = *$, then $g(r(s)) = (\mathbf{F}r)(f(s)) = *$ (thus s is a leaf iff $r(s)$ is one),
2. $f(s) = \langle s_1, s_2 \rangle$, then $g(r(s)) = \langle t_1, t_2 \rangle$ with $t_1 = r(s_1)$ and $t_2 = r(s_2)$ (thus $r(s)$ branches out to $\langle r(s_1), r(s_2) \rangle$, provided s branches out to $\langle s_1, s_2 \rangle$).

Thus a coalgebra morphism preserves the tree structure. \mathbb{M}

Example 1.134 Continuing the discussion of deterministic automata with output from Example 1.127, let (S, f) and (T, g) be \mathbf{F} -coalgebras and $r : (S, f) \rightarrow (T, g)$ be a morphism. Given state $s \in S$, let $f(s)(a) = \langle s', b \rangle$ be the new state and the output, respectively, after input $a \in A$ for automaton (S, f) . Then $g(r(s))(a) = \langle r(s'), b \rangle$, so after input $a \in A$ the automaton (T, g) will be in state $r(s)$ and give the output b , as expected. Hence coalgebra morphisms preserve the automatas' working. \mathbb{M}

Example 1.135 Continuing the the discussion of transition systems from Example 1.126, let (S, f) and (T, g) be labelled transition systems with A as the set of actions. Thus a transition

from s to s' on action a is given in (S, f) iff $\langle a, s' \rangle \in f(s)$. Let us just for convenience write $s \rightsquigarrow_{a,S} s'$ iff this is the case, similarly, we write $t \rightsquigarrow_{a,T} t'$ iff $t, t' \in T$ with $\langle a, t' \rangle \in g(t)$.

Now let $r : (S, f) \rightarrow (T, g)$ be a coalgebra morphism. We claim that for given $s \in S$ we have a transition $r(s) \rightsquigarrow_{a,T} t_0$ for some t_0 iff we can find s_0 such that $s \rightsquigarrow_{a,S} s_0$ and $r(s_0) = t_0$. Because $r : (S, f) \rightarrow (T, g)$ is a coalgebra morphism, we have $g \circ r = (\mathbf{T}r) \circ f$ with $\mathbf{T} = \mathcal{P}(A \times -)$. Thus

$$g(r(s)) = \mathcal{P}(A \times r)(s) = \{\langle a, r(s') \rangle \mid \langle a, s' \rangle \in f(s)\}.$$

Consequently,

$$\begin{aligned} r(s) \rightsquigarrow_{a,T} t_0 &\Leftrightarrow \langle a, t_0 \rangle \in g(r(s)) \\ &\Leftrightarrow \langle a, t_0 \rangle = \langle a, r(s_0) \rangle \text{ for some } \langle a, s_0 \rangle \in f(s) \\ &\Leftrightarrow s \rightsquigarrow_{a,S} s_0 \text{ for some } s_0 \text{ with } r(s_0) = t_0 \end{aligned}$$

This means that the transitions in (T, g) are essentially controlled by the morphism r and the transitions in (S, f) . Hence a coalgebra morphism between transition systems is a bounded morphism in the sense of Example 1.10. \upharpoonright

Example 1.136 We continue the discussion of upper closed sets from Example 1.131. Let (S, f) and (T, g) be \mathbf{V} -coalgebras, so this diagram is commutative for morphism $r : (S, F) \rightarrow (T, g)$:

$$\begin{array}{ccc} S & \xrightarrow{r} & T \\ f \downarrow & & \downarrow g \\ \mathbf{V}S & \xrightarrow{\mathbf{V}r} & \mathbf{V}T \end{array}$$

Consequently, $W \in g(r(s))$ iff $r^{-1}[W] \in f(s)$. Taking up the interpretation of sets of states which may be achieved by a player, we see that it¹ may achieve W in state $r(s)$ in (T, g) iff it may achieve in (S, f) the set $r^{-1}[W]$ in state s . \upharpoonright

1.6.1 Bisimulations

The notion of bisimilarity is fundamental for the application of coalgebras to system modelling. Bisimilar coalgebras behave in a similar fashion, witnessed by a mediating system.

Definition 1.137 Let \mathbf{F} be an endofunctor on a category \mathbf{K} . The \mathbf{F} -coalgebras (S, f) and (T, g) are said to be bisimilar iff there exists a coalgebra (M, m) and coalgebra morphisms $(S, f) \longleftarrow (M, m) \longrightarrow (T, g)$. The coalgebra (M, m) is called mediating.

Thus we obtain this characteristic diagram with ℓ and r as the corresponding morphisms.

$$\begin{array}{ccccc} S & \xleftarrow{\ell} & M & \xrightarrow{r} & T \\ f \downarrow & & \downarrow m & & \downarrow g \\ \mathbf{F}S & \xleftarrow{\mathbf{F}\ell} & \mathbf{F}M & \xrightarrow{\mathbf{F}r} & \mathbf{F}T \end{array}$$

¹The present author is not really sure about the players' gender — players are female in the overwhelming majority of papers in the literature, but on the other hand are addressed as *Angel* or *Demon*; this may be politically correct, but does not seem to be biblically so with a view toward Matthew 22:30. To be on the safe side — it is so hopeless to argue with feminists — players are neutral in the present treatise.

Thus we have

$$\begin{aligned} f \circ \ell &= (\mathbf{F}\ell) \circ \mathbf{m} \\ g \circ r &= (\mathbf{F}r) \circ \mathbf{m} \end{aligned}$$

In this way it is easy to see why (M, \mathbf{m}) is called mediating.

Bisimilarity was originally investigated when concurrent systems became of interest. The original formulation, however, was not coalgebraic but rather relational.

Definition 1.138 *Let (S, \rightsquigarrow_S) and (T, \rightsquigarrow_T) be transition systems. Then $B \subseteq S \times T$ is called a bisimulation iff for all $\langle s, t \rangle \in B$ these conditions are satisfied:*

1. *if $s \rightsquigarrow_S s'$, then there is a $t' \in T$ such that $t \rightsquigarrow_T t'$ and $\langle s', t' \rangle \in B$.*
2. *if $t \rightsquigarrow_T t'$, then there is a $s' \in S$ such that $s \rightsquigarrow_S s'$ and $\langle s', t' \rangle \in B$.*

Hence a bisimulation simulates transitions in one system through the other one. On first sight, these notions of bisimilarity are not related to each other. Recall that transition systems are coalgebras for the power set functor \mathcal{P} . This is the connection:

Theorem 1.139 *Given the transition systems (S, \rightsquigarrow_S) and (T, \rightsquigarrow_T) with the associated \mathcal{P} -coalgebras (S, f) and (T, g) , then these statements are equivalent for $B \subseteq S \times T$:*

1. *B is a bisimulation.*
2. *There exists a \mathcal{P} -coalgebra structure h on B such that $(S, f) \longleftarrow (B, h) \longrightarrow (T, g)$ with the projections as morphisms is mediating.*

Proof That $(S, f) \xleftarrow{\pi_S} (B, h) \xrightarrow{\pi_T} (T, g)$ is mediating follows from commutativity of this diagram.

$$\begin{array}{ccccc} S & \xleftarrow{\pi_S} & B & \xrightarrow{\pi_T} & T \\ f \downarrow & & h \downarrow & & \downarrow g \\ \mathcal{P}(S) & \xleftarrow{\mathcal{P}(\pi_S)} & \mathcal{P}(B) & \xrightarrow{\mathcal{P}(\pi_T)} & \mathcal{P}(T) \end{array}$$

1 \Rightarrow 2: We have to construct a map $h : B \rightarrow \mathcal{P}(B)$ such that

$$\begin{aligned} f(\pi_S(s, t)) &= \mathcal{P}(\pi_S)(h(s, t)) \\ f(\pi_T(s, t)) &= \mathcal{P}(\pi_T)(h(s, t)) \end{aligned}$$

for all $\langle s, t \rangle \in B$. The choice is somewhat obvious: put for $\langle s, t \rangle \in B$

$$h(s, t) := \{\langle s', t' \rangle \mid s \rightsquigarrow_S s', t \rightsquigarrow_T t'\}.$$

Thus $h : B \rightarrow \mathcal{P}(B)$ is a map, hence (B, h) is a \mathcal{P} -coalgebra.

Now fix $\langle s, t \rangle \in B$, then we claim that $f(s) = \mathcal{P}(\pi_S)(h(s, t))$.

“ \subseteq ”: Let $s' \in f(s)$, hence $s \rightsquigarrow_S s'$, thus there exists t' with $\langle s', t' \rangle \in B$ such that $t \rightsquigarrow_T t'$, hence

$$\begin{aligned} s' &\in \{\pi_S(s_0, t_0) \mid \langle s_0, t_0 \rangle \in h(s, t)\} \\ &= \{s_0 \mid \langle s_0, t_0 \rangle \in h(s, t) \text{ for some } t_0\} \\ &= \mathcal{P}(\pi_S)(h(s, t)). \end{aligned}$$

“ \supseteq ” If $s' \in \mathcal{P}(\pi_S)(h(s, t))$, then in particular $s \rightsquigarrow_S s'$, thus $s' \in f(s)$.

Thus we have shown that $\mathcal{P}(\pi_S)(h(s, t)) = f(s) = f(\pi_S(s, t))$. One shows $\mathcal{P}(\pi_T)(h(s, t)) = g(t) = f(\pi_T(s, t))$ in exactly the same way. We have constructed h such that (B, h) is a \mathcal{P} -coalgebra, and such that the diagrams above commute.

2 \Rightarrow 1: Assume that h exists with the properties described in the assertion, then we have to show that B is a bisimulation. Now let $\langle s, t \rangle \in B$ and $s \rightsquigarrow_S s'$, hence $s' \in f(s) = f(\pi_S(s, t)) = \mathcal{P}(\pi_S)(h(s, t))$. Thus there exists t' with $\langle s', t' \rangle \in h(s, t) \subseteq B$, and hence $\langle s', t' \rangle \in B$. We claim that $t \rightsquigarrow_T t'$, which is tantamount to saying $t' \in g(t)$. But $g(t) = \mathcal{P}(\pi_T)(h(s, t))$, and $\langle s', t' \rangle \in h(s, t)$, hence $t' \in \mathcal{P}(\pi_T)(h(s, t)) = g(t)$. This establishes $t \rightsquigarrow_T t'$. A similar argument finds s' with $s \rightsquigarrow_S s'$ with $\langle s', t' \rangle \in B$ in case $t \rightsquigarrow_T t'$.

This completes the proof. \dashv

Thus for transition systems we may use bisimulations as relations and bisimulations as coalgebras interchangeably. The connection to \mathcal{P} -coalgebra morphisms and bisimulations is further strengthened by investigating the graph of a morphism (recall that the graph $\text{Graph}(r)$ of a map $r : S \rightarrow T$ is the relation $\{\langle s, r(s) \rangle \mid s \in S\}$).

Proposition 1.140 *Given coalgebras (S, f) and (T, g) for the power set functor \mathcal{P} , $r : (S, f) \rightarrow (T, g)$ is a morphism iff $\text{Graph}(r)$ is a bisimulation for (S, f) and (T, g) .*

Proof 1. Assume that $r : (S, f) \rightarrow (T, g)$ is a morphism, so that $g \circ r = \mathcal{P}(r) \circ f$. Now define

$$h(s, t) := \{\langle s', r(s') \rangle \mid s' \in f(s)\} \subseteq \text{Graph}(r)$$

for $\langle s, t \rangle \in \text{Graph}(r)$. Then $g(\pi_T(s, t)) = g(t) = \mathcal{P}(\pi_T)(h(s, t))$ for $t = r(s)$.

“ \subseteq ” If $t' \in g(t)$ for $t = r(s)$, then

$$\begin{aligned} t' \in g(r(s)) &= \mathcal{P}(r)(f(s)) \\ &= \{r(s') \mid s' \in f(s)\} \\ &= \mathcal{P}(\pi_T)(\{\langle s', r(s') \rangle \mid s' \in f(s)\}) \\ &= \mathcal{P}(\pi_T)(h(s, t)) \end{aligned}$$

“ \supseteq ” If $\langle s', t' \rangle \in h(s, t)$, then $s' \in f(s)$ and $t' = r(s')$, but this implies $t' \in \mathcal{P}(r)(f(s)) = g(r(s))$.

Thus $g \circ \pi_T = \mathcal{P}(\pi_T) \circ h$. The equation $f \circ \pi_S = \mathcal{P}(\pi_S) \circ h$ is established similarly.

Hence we have found a coalgebra structure h on $\text{Graph}(r)$ such that

$$(S, f) \xleftarrow{\pi_S} (\text{Graph}(r), h) \xrightarrow{\pi_T} (T, g)$$

are coalgebra morphisms, so that $(\text{Graph}(r), h)$ is a bisimulation.

2. If, conversely, $(\text{Graph}(r), h)$ is a bisimulation with the projections as morphisms, then we have $r = \pi_T \circ \pi_S^{-1}$. Then π_T is a morphism, and π_S^{-1} is a morphism as well (note that we work on the graph of r). So r is a morphism. \dashv

Let us have a look at the situation with the upper closed sets from Example 1.101. There we find a comparable situation.

Definition 1.141 *Let*

$$\mathbf{VS} := \{V \subseteq \mathcal{P}(S) \mid V \text{ is upper closed}\}$$

*be the endofunctor on **Set** which assigns to set S all upper closed subsets of $\mathcal{P}S$. Given \mathbf{V} -coalgebras (S, f) and (T, g) , a subset $B \subseteq S \times T$ is called a bisimulation of (S, f) and (T, g) iff for each $\langle s, t \rangle \in B$*

1. *For all $X \in f(s)$ there exists $Y \in g(t)$ such that for each $t' \in Y$ there exists $s' \in X$ with $\langle s', t' \rangle \in B$.*
2. *For all $Y \in g(t)$ there exists $X \in f(s)$ such that for each $s' \in X$ there exists $t' \in Y$ with $\langle s', t' \rangle \in B$.*

We have then a comparable characterization of bisimilar coalgebras.

Proposition 1.142 *Let (S, f) and (T, g) be coalgebras for \mathbf{V} . Then the following statements are equivalent for $B \subseteq S \times T$ with $\pi_S[B] = S$ and $\pi_T[B] = T$*

1. *B is a bisimulation of (S, f) and (T, g) .*
2. *There exists a coalgebra structure h on B so that the projections $\pi_S : B \rightarrow S, \pi_T : B \rightarrow T$ are morphisms $(S, f) \xleftarrow{\pi_S} (B, h) \xrightarrow{\pi_T} (T, g)$.*

Proof 1 \Rightarrow 2: Define

$$h(s, t) := \{D \subseteq B \mid \pi_S[D] \in f(s) \text{ and } \pi_T[D] \in g(t)\},$$

$\langle s, t \rangle \in B$. Hence $h(s, t) \subseteq \mathcal{P}(B)$, and because both $f(s)$ and $g(t)$ are upper closed, so is $h(s, t)$.

Now fix $\langle s, t \rangle \in B$. We show first that $f(s) = \{\pi_S[Z] \mid Z \in h(s, t)\}$. From the definition of $h(s, t)$ it follows that $\pi_S[Z] \in f(s)$ for each $Z \in h(s, t)$. So we have to establish the other inclusion. Let $X \in f(s)$, then $X = \pi_S[\pi_S^{-1}[X]]$, because $\pi_S : B \rightarrow S$ is onto, so it suffices to show that $\pi_S^{-1}[X] \in h(s, t)$, hence that $\pi_T[\pi_S^{-1}[X]] \in g(t)$. Given X there exists $Y \in g(t)$ so that for each $t' \in Y$ there exists $s' \in X$ such that $\langle s', t' \rangle \in B$. Thus $Y = \pi_T[(X \times Y) \cap B]$. But this implies $Y \subseteq \pi_T[\pi_S^{-1}[X]]$, hence $Y \subseteq \pi_T[\pi_S^{-1}[X]] \in g(t)$. One similarly shows that $g(t) = \{\pi_T[Z] \mid Z \in h(s, t)\}$.

In a second step, we show that $\{\pi_S[Z] \mid Z \in h(s, t)\} = \{C \mid \pi_S^{-1}[C] \in h(s, t)\}$. In fact, if $C = \pi_S[Z]$ for some $Z \in h(s, t)$, then $Z \subseteq \pi_S^{-1}[C] = \pi_S^{-1}[\pi_S[Z]]$, hence $\pi_S^{-1}[C] \in h(s, t)$. If, conversely, $Z := \pi_S^{-1}[C] \in h(s, t)$, then $C = \pi_S[Z]$. Thus we obtain for $\langle s, t \rangle \in B$

$$\begin{aligned} f(s) &= \{\pi_S[Z] \mid Z \in h(s, t)\} \\ &= \{C \mid \pi_S^{-1}[C] \in h(s, t)\} \\ &= (\mathbf{V}\pi_S)(h(s, t)). \end{aligned}$$

Summarizing, this means that $\pi_S : (B, h) \rightarrow (S, f)$ is a morphism. A very similar proof shows that $\pi_T : (B, h) \rightarrow (T, g)$ is a morphism as well.

2 \Rightarrow 1: Now assume that the projections are coalgebra morphisms, and let $\langle s, t \rangle \in B$. Given $X \in f(s)$, we know that $X = \pi_S[Z]$ for some $Z \in h(s, t)$. Thus we find for any $t' \in Y$ some $s' \in X$ with $\langle s', t' \rangle \in B$. The symmetric property of a bisimulation is proved exactly in the same way. Hence B is a bisimulation for (S, f) and (T, g) . \dashv

Encouraged by these observations, we define bisimulations for set based functors, i.e., for endofunctors on the category **Set** of sets with maps as morphisms. This is nothing but a specialization of the general notion of bisimilarity, taking specifically into account that we may in **Set** to consider subsets of the Cartesian product, and that we have projections at our disposal.

Definition 1.143 *Let \mathbf{F} be an endofunctor on **Set**. Then $R \subseteq S \times T$ is called a bisimulation for the \mathbf{F} -coalgebras (S, f) and (T, g) iff there exists a map $h : R \rightarrow \mathbf{F}(R)$ rendering this diagram commutative:*

$$\begin{array}{ccccc} S & \xleftarrow{\pi_S} & R & \xrightarrow{\pi_T} & T \\ f \downarrow & & h \downarrow & & \downarrow g \\ \mathbf{F}(S) & \xleftarrow{\mathbf{F}\pi_S} & \mathbf{F}(R) & \xrightarrow{\mathbf{F}\pi_T} & \mathbf{F}(T) \end{array}$$

These are immediate consequences:

Lemma 1.144 $\Delta_S := \{\langle s, s \rangle \mid s \in S\}$ is a bisimulation for every \mathbf{F} -coalgebra (S, f) . If R is a bisimulation for the \mathbf{F} -coalgebras (S, f) and (T, g) , then R^{-1} is a bisimulation for (T, g) and (S, f) . \dashv

It is instructive to look back and investigate again the graph of a morphism $r : (S, f) \rightarrow (T, g)$, where this time we do not have the power set functor — as in Proposition 1.140 — but a general endofunctor \mathbf{F} on **Set**.

Corollary 1.145 *Given coalgebras (S, f) and (T, g) for the endofunctor \mathbf{F} on **Set**, $r : (S, f) \rightarrow (T, g)$ is a morphism iff $\text{Graph}(r)$ is a bisimulation for (S, f) and (T, g) .*

Proof 0. The proof for Proposition 1.140 needs some small adjustments, because we do not know how exactly functor \mathbf{F} is operating on maps.

1. If $r : (S, f) \rightarrow (T, g)$ is a morphism, we know that $g \circ r = \mathbf{F}(r) \circ f$. Consider the map $\tau : S \rightarrow S \times T$ which is defined as $s \mapsto \langle s, r(s) \rangle$, thus $\mathbf{F}(\tau) : \mathbf{F}(S) \rightarrow \mathbf{F}(S \times T)$. Define

$$h : \begin{cases} \text{Graph}(r) & \rightarrow \mathbf{F}(\text{Graph}(r)) \\ \langle s, r(s) \rangle & \mapsto \mathbf{F}(\tau)(f(s)) \end{cases}$$

Then it is not difficult to see that both $g \circ \pi_T = \mathbf{F}(\pi_T) \circ h$ and $f \circ \pi_S = \mathbf{F}(\pi_S) \circ h$ holds. Hence $(\text{Graph}(r), h)$ is an \mathbf{F} -coalgebra mediating between (S, f) and (T, g) .

2. Assume that $\text{Graph}(r)$ is a bisimulation for (S, f) and (T, g) , then both π_T and π_S^{-1} are morphisms for the \mathbf{F} -coalgebras, so the proof proceeds exactly as the corresponding one for Proposition 1.140. \dashv

We will study some properties of bisimulations now, including a preservation property of functor $\mathbf{F} : \mathbf{Set} \rightarrow \mathbf{Set}$. This functor will be fixed for the time being.

We may construct bisimulations from morphisms.

Lemma 1.146 *Let (S, f) , (T, g) and (U, h) be \mathbf{F} -coalgebras with morphisms $\varphi : (S, f) \rightarrow (T, g)$ and $\psi : (S, f) \rightarrow (U, h)$. Then the image of S under $\varphi \times \psi$,*

$$\langle \varphi, \psi \rangle[S] := \{\langle \varphi(s), \psi(s) \rangle \mid s \in S\}$$

is a bisimulation for (T, g) and (U, h) .

Proof 0. Look at this diagram

$$\begin{array}{ccccc}
 & & \langle \varphi, \psi \rangle [S] & & \\
 & \swarrow \pi_T & \uparrow j & \searrow \pi_U & \\
 T & & S & & U \\
 & \swarrow \varphi & & \searrow \psi & \\
 & & & &
 \end{array}$$

Here $j(s) := \langle \varphi(s), \psi(s) \rangle$, hence $j : S \rightarrow \langle \varphi, \psi \rangle [S]$ is surjective. We can find a map $i : \langle \varphi, \psi \rangle [S] \rightarrow S$ so that $j \circ i = \text{id}_{\langle \varphi, \psi \rangle [S]}$ using the Axiom of Choice². So we have a left inverse to j , which will help us in the construction below.

1. We want to define a coalgebra structure for $\langle \varphi, \psi \rangle [S]$ such that the diagram below commutes, i.e., forms a bisimulation diagram. Put $k := \mathbf{F}(j) \circ f \circ i$, then we have

$$\begin{array}{ccccc}
 T & \xleftarrow{\pi_T} & \langle \varphi, \psi \rangle [S] & \xrightarrow{\pi_U} & U \\
 g \downarrow & & \downarrow k & & \downarrow h \\
 \mathbf{F}T & \xleftarrow{\mathbf{F}\pi_T} & \mathbf{F}(\langle \varphi, \psi \rangle [S]) & \xrightarrow{\mathbf{F}\pi_U} & \mathbf{F}U
 \end{array}$$

Now

$$\begin{aligned}
 \mathbf{F}(\pi_T) \circ k &= \mathbf{F}(\pi_T) \circ \mathbf{F}(j) \circ f \circ i \\
 &= \mathbf{F}(\pi_T \circ j) \circ f \circ i \\
 &= \mathbf{F}(\varphi) \circ f \circ i && (\text{since } \pi_T \circ j = \varphi) \\
 &= g \circ \varphi \circ i && (\text{since } \mathbf{F}(\varphi) \circ f = g \circ \varphi) \\
 &= g \circ \pi_T
 \end{aligned}$$

Hence the left hand diagram commutes. Similarly

$$\begin{aligned}
 \mathbf{F}(\pi_U) \circ k &= \mathbf{F}(\pi_U \circ j) \circ f \circ i \\
 &= \mathbf{F}(\psi) \circ f \circ i \\
 &= h \circ \psi \circ i \\
 &= h \circ \pi_U
 \end{aligned}$$

Thus we obtain a commutative diagram on the right hand as well. \dashv

This is applied to the composition of relations:

Lemma 1.147 *Let $R \subseteq S \times T$ and $Q \subseteq T \times U$ be relations, and put $X := \{\langle s, t, u \rangle \mid \langle s, t \rangle \in R, \langle t, u \rangle \in Q\}$. Then*

$$R \circ Q = \langle \pi_S \circ \pi_R, \pi_U \circ \pi_Q \rangle [X].$$

²For each $r \in \langle \varphi, \psi \rangle [S]$ there exists at least one $s \in S$ with $r = \langle \varphi(s), \psi(s) \rangle$. Pick for each r such an s and call it $i(r)$, thus $r = \langle \varphi(i(r)), \psi(i(r)) \rangle$.

Proof Simply trace an element of $R \circ Q$ through this construction:

$$\begin{aligned} \langle s, u \rangle \in R &\Leftrightarrow \exists t \in T : \langle s, t \rangle \in R, \langle t, u \rangle \in Q \\ &\Leftrightarrow \exists t \in T : \langle s, t, u \rangle \in X \\ &\Leftrightarrow \exists t \in T : s = (\pi_S \circ \pi_R)(s, t, u) \text{ and } u = (\pi_U \circ \pi_Q)(s, t, u). \end{aligned}$$

⊥

Looking at X in its relation to the projections, we see that X is actually a weak pullback, to be precise:

Lemma 1.148 *Let R, Q, X be as above, then X is a weak pullback of $\pi_T^R : R \rightarrow T$ and $\pi_T^Q : Q \rightarrow T$, so that in particular $\pi_T^Q \circ \pi_Q = \pi_T^R \circ \pi_R$.*

Proof 1. It is easy to see that this diagram

$$\begin{array}{ccc} X & \xrightarrow{\pi_Q} & Q \\ \pi_R \downarrow & & \downarrow \pi_Q^T \\ R & \xrightarrow{\pi_T^R} & T \end{array}$$

commutes. In fact, given $\langle s, t, u \rangle \in X$, we know that $\langle s, t \rangle \in R$ and $\langle t, u \rangle \in Q$, hence $(\pi_T^Q \circ \pi_Q)(s, t, u) = \pi_T^Q(t, u) = t$ and $(\pi_T^R \circ \pi_R)(s, t, u) = \pi_T^R(s, t) = t$.

2. If $f_1 : Y \rightarrow R$ and $f_2 : Y \rightarrow Q$ are maps for some set Y such that $\pi_T^R \circ f_1 = \pi_T^R \circ f_2$, we can write $f_1(y) = \langle f_1^S(y), f_1^T(y) \rangle \in R$ and $f_2(y) = \langle f_2^T(y), f_2^U(y) \rangle \in Q$. Put $\sigma(y) := \langle f_1^S(y), f_2^T(y), f_2^U(y) \rangle$, then $\sigma : Y \rightarrow X$ with $f_1 = \pi_R \circ \sigma$ and $f_2 = \pi_Q \circ \sigma$. Thus X is a weak pullback. ⊥

It will turn out that the functor should preserve the pullback property. Preserving the uniqueness property of a pullback will be too strong a requirement, but preserving weak pullbacks will be helpful and not too restrictive.

Definition 1.149 *Functor F preserves weak pullbacks iff F maps weak pullbacks to weak pullbacks.*

Thus a weak pullback diagram like

$$\begin{array}{ccc} H & \xrightarrow{g'} & X \\ \downarrow f' & \searrow & \downarrow h \\ P & \xrightarrow{g} & X \\ \downarrow f & & \downarrow h \\ Y & \xrightarrow{i} & Z \end{array} \quad \text{translates to} \quad \begin{array}{ccc} H & \xrightarrow{g'} & FX \\ \downarrow f' & \searrow & \downarrow Fh \\ FP & \xrightarrow{Fg} & FX \\ \downarrow Ff & & \downarrow Fh \\ FY & \xrightarrow{Fi} & FZ \end{array}$$

We want to show that the composition of bisimulations is a bisimulation again: this requires that the functor preserves weak pullbacks. Before we state and prove a corresponding property, we need an auxiliary statement which is of independent interest, viz., that the weak pullback of bisimulations forms a bisimulation again. To be specific:

Lemma 1.150 *Assume that functor \mathbf{F} preserves weak pullbacks, and let $r : (S, f) \rightarrow (T, g)$ and $s : (U, h) \rightarrow (T, g)$ be morphisms for the \mathbf{F} -coalgebras (S, f) , (T, g) and (U, h) . Then there exists a coalgebra structure $p : P \rightarrow \mathbf{F}P$ for the weak pullback P of r and s with projections π_S and π_U such that (P, p) is a bismulation for (S, f) and (U, h) .*

Proof We will need these diagrams

$$\begin{array}{ccccc} S & \xrightarrow{r} & R & \xleftarrow{s} & U \\ f \downarrow & & g \downarrow & & \downarrow h \\ \mathbf{F}S & \xrightarrow{\mathbf{F}r} & \mathbf{F}T & \xleftarrow{\mathbf{F}s} & \mathbf{F}U \end{array} \quad (12)$$

$$\begin{array}{ccc} P & \xrightarrow{\pi_U} & U \\ \pi_S \downarrow & & \downarrow s \\ S & \xrightarrow{r} & T \end{array} \quad (13)$$

$$\begin{array}{ccccc} S & \xleftarrow{\pi_S} & P & \xrightarrow{\pi_U} & U \\ f \downarrow & & \downarrow ? \downarrow ? & & \downarrow h \\ \mathbf{F}S & \xleftarrow{\mathbf{F}\pi_S} & \mathbf{F}P & \xrightarrow{\mathbf{F}\pi_U} & \mathbf{F}U \end{array} \quad (14)$$

While the first two diagrams are helping with the proof's argument, the third diagram has a gap in the middle. We want to find an arrow $P \rightarrow \mathbf{F}P$ so that the diagrams will commute. Actually, the weak pullback will help us obtaining this information.

Because

$$\begin{aligned} \mathbf{F}(r) \circ f \circ \pi_S &= g \circ r \circ \pi_S && \text{(diagram 12, left)} \\ &= g \circ s \circ \pi_U && \text{(diagram 13)} \\ &= \mathbf{F}(s) \circ h \circ \pi_U && \text{(diagram 12, right)} \end{aligned}$$

we may conclude that $\mathbf{F}(r) \circ f \circ \pi_S = \mathbf{F}(s) \circ h \circ \pi_U$. Diagram 13 is a pullback diagram. Because \mathbf{F} preserves weak pullbacks, this diagram can be complemented by an arrow $P \rightarrow \mathbf{F}P$ rendering the triangles commutative.

$$\begin{array}{ccccc} P & & & & \\ & \searrow h \circ \pi_U & & & \\ & & \mathbf{F}P & \xrightarrow{\mathbf{F}\pi_U} & \mathbf{F}U \\ & \searrow f \circ \pi_S & \downarrow \mathbf{F}\pi_S & & \downarrow \mathbf{F}s \\ & & \mathbf{F}S & \xrightarrow{\mathbf{F}r} & \mathbf{F}T \end{array}$$

Hence there exists $p : P \rightarrow \mathbf{F}P$ with $\mathbf{F}(\pi_S) \circ p = f \circ \pi_S$ and $\mathbf{F}(\pi_U) \circ p = h \circ \pi_U$. Thus p makes diagram (14) a bismulation diagram. \dashv

Now we are in a position to show that the composition of bismulations is a bismulation again, provided the functor \mathbf{F} behaves decently.

Proposition 1.151 *Let R be a bisimulation of (S, f) and (T, g) , Q be a bisimulation of (T, g) and (U, h) , and assume that \mathbf{F} preserves weak pullbacks. Then $R \circ Q$ is a bisimulation of (S, f) and (U, h) .*

Proof We can write $R \circ Q = \langle \pi_S \circ \pi_R, \pi_U \circ \pi_Q \rangle [X]$ with $X := \{ \langle s, t, u \rangle \mid \langle s, t \rangle \in R, \langle t, u \rangle \in Q \}$. Since X is a weak pullback of π_T^R and π_T^Q by Lemma 1.148, we know that X is a bisimulation of (R, r) and (Q, q) , with r and q as the dynamics of the corresponding \mathbf{F} -coalgebras. $\pi_S \circ \pi_R : X \rightarrow S$ and $\pi_U \circ \pi_Q : X \rightarrow U$ are morphisms, thus $\langle \pi_S \circ \pi_R, \pi_U \circ \pi_Q \rangle [X]$ is a bisimulation, since X is a weak pullback. Thus the assertion follows from Lemma 1.147. \dashv

The proof shows in which way the existence of the morphism $P \rightarrow \mathbf{F}P$ is used for achieving the desired properties.

Let us have a look at bisimulations on a coalgebra. Here bisimulations may have an additional structure, viz., they may be equivalence relations as well. Accordingly, we call these bisimulations *bisimulation equivalences*. Hence given a coalgebra (S, f) , a bisimulation equivalence α for (S, f) is a bisimulation for (S, f) which is also an equivalence relation. While bisimulations carry properties which are concerned with the coalgebraic structure, an equivalence relation is purely related to the set structure. It is, however, fairly natural to ask in view of the properties which we did explore so far (Lemma 1.144, Proposition 1.151) whether or not we can take a bisimulation and turn it into an equivalence relation, or at least do so under favorable conditions on functor \mathbf{F} . We will deal with this question and some of its cousins now.

Observe first that the factor space of a bisimulation equivalence can be turned into a coalgebra.

Lemma 1.152 *Let (S, f) be an \mathbf{F} -coalgebra, and α be a bisimulation equivalence on (S, f) . Then there exists a unique dynamics $\alpha_R : S/\alpha \rightarrow \mathbf{F}(S/\alpha)$ with $\mathbf{F}(\eta_\alpha) \circ f = \alpha_R \circ \eta_\alpha$.*

Proof Because α is in particular a bisimulation, we know that there exists by Theorem 1.139 a dynamics $\rho : \alpha \rightarrow \mathbf{F}\alpha$ rendering this diagram commutative.

$$\begin{array}{ccccc}
 S & \xleftarrow{\pi_S^{(1)}} & \alpha & \xrightarrow{\pi_S^{(2)}} & S \\
 f \downarrow & & \downarrow \rho & & \downarrow f \\
 \mathbf{F}S & \xleftarrow{\mathbf{F}\pi_S^{(1)}} & \mathbf{F}\alpha & \xrightarrow{\mathbf{F}\pi_S^{(2)}} & \mathbf{F}S
 \end{array}$$

The obvious choice would be to set $\alpha_R([s]_\alpha) := (\mathbf{F}(\eta_\alpha) \circ f)(s)$, but this is only possible if we know that the map is well defined, so we have to check whether $(\mathbf{F}(\eta_\alpha) \circ f)(s_1) = (\mathbf{F}(\eta_\alpha) \circ f)(s_2)$ holds, whenever $s_1 \alpha s_2$.

In fact, $s_1 \alpha s_2$ means $\langle s_1, s_2 \rangle \in \alpha$, so that $f(s_1) = f(\pi_S^{(1)}(s_1, s_2)) = (\mathbf{F}(\pi_S^{(1)}) \circ \rho)(s_1, s_2)$, similarly for $f(s_2)$. Because α is an equivalence relation, we have $\eta_\alpha \circ \pi_S^{(1)} = \eta_\alpha \circ \pi_S^{(2)}$. Thus

$$\begin{aligned}
 \mathbf{F}(\eta_\alpha)(f(s_1)) &= (\mathbf{F}(\eta_\alpha \circ \pi_S^{(1)}) \circ \rho)(s_1, s_2) \\
 &= (\mathbf{F}(\eta_\alpha \circ \pi_S^{(2)}) \circ \rho)(s_1, s_2) \\
 &= \mathbf{F}(\eta_\alpha)(f(s_2))
 \end{aligned}$$

This means that α_R is well defined indeed, and that η_α is a morphism. Hence the dynamics α_R exists and renders η_α a morphism.

Now assume that $\beta_R : S/\alpha \rightarrow \mathbf{F}(S/\alpha)$ satisfies also $\mathbf{F}(\eta_\alpha) \circ f = \beta_R \circ \eta_\alpha$. But then $\beta_R \circ \eta_\alpha = \mathbf{F}(\eta_\alpha) \circ f = \alpha_R \circ \eta_\alpha$, and, since η_α is onto, it is an epi, so that we may conclude $\beta_R = \alpha_R$. Hence α_R is uniquely determined. \dashv

Bisimulations can be transported along morphisms, if the functor preserves weak pullbacks.

Proposition 1.153 *Assume that \mathbf{F} preserves weak pullbacks, and let $r : (S, f) \rightarrow (T, g)$ be a morphisms. Then*

1. *If R is a bisimulation on (S, f) , then $(r \times r)[R] = \{\langle r(s), r(s') \rangle \mid \langle s, s' \rangle \in R\}$ is a bisimulation on (T, g) .*
2. *If Q is a bisimulation on (T, g) , then $(r \times r)^{-1}[Q] = \{\langle s, s' \rangle \mid \langle r(s), r(s') \rangle \in Q\}$ is a bisimulation on (S, f) .*

Proof 0. Note that $\text{Graph}(r)$ is a bisimulation by Corollary 1.145, because r is a morphism.

1. We claim that

$$(r \times r)[R] = (\text{Graph}(r))^{-1} \circ R \circ \text{Graph}(r)$$

holds. Granted that, we can apply Proposition 1.151 together with Lemma 1.144 for establishing the first property. But $\langle t, t' \rangle \in (r \times r)[R]$ iff we can find $\langle s, s' \rangle \in R$ with $\langle t, t' \rangle = \langle r(s), r(s') \rangle$, hence $\langle r(s), s \rangle \in \text{Graph}(r)^{-1}$, $\langle s, s' \rangle \in R$ and $\langle s', r(s') \rangle \in \text{Graph}(r)$, hence iff $\langle t, t' \rangle \in \text{Graph}(r)^{-1} \circ R \circ \text{Graph}(r)$.

2. Similarly, we show that $(r \times r)^{-1}[Q] = \text{Graph}(r) \circ R \circ \text{Graph}(r)^{-1}$. This is left to the reader. \dashv

For investigating further structural properties, we need

Lemma 1.154 *If (S, f) and (T, g) are \mathbf{F} -coalgebras, then there exists a unique coalgebraic structure on $S + T$ such that the injections i_S and i_T are morphisms.*

Proof We have to find a morphism $S + T \rightarrow \mathbf{F}(S + T)$ such that this diagram is commutative

$$\begin{array}{ccccc} S & \xrightarrow{i_S} & S + T & \xleftarrow{i_T} & T \\ f \downarrow & & \downarrow & & \downarrow g \\ \mathbf{F}S & \xrightarrow{\mathbf{F}i_S} & \mathbf{F}(S + T) & \xleftarrow{\mathbf{F}i_T} & \mathbf{F}T \end{array}$$

Because $\mathbf{F}(i_S) \circ f : S \rightarrow \mathbf{F}(S + T)$ and $\mathbf{F}(i_T) \circ g : T \rightarrow \mathbf{F}(S + T)$ are morphisms, there exists a unique morphism $h : S + T \rightarrow \mathbf{F}(S + T)$ with $h \circ i_S = \mathbf{F}(i_S) \circ f$ and $h \circ i_T = \mathbf{F}(i_T) \circ g$. Thus $(S + T, h)$ is a coalgebra, and i_S as well as i_T are morphisms. \dashv

The attempt to establish a comparable property for the product could not work with the universal property for products, as a look at the universal property for products will show.

We obtain as a consequence that bisimulations are closed under finite unions.

Lemma 1.155 *Let (S, f) and (T, g) be coalgebras with bisimulations R_1 and R_2 . Then $R_1 \cup R_2$ is a bisimulation.*

Proof 1. We can find morphisms $r_i : R_i \rightarrow \mathbf{F}R_i$ for $i = 1, 2$ rendering the corresponding bisimulation diagrams commutative. Then $R_1 + R_2$ is an \mathbf{F} -coalgebra with

$$\begin{array}{ccccc} R_1 & \xrightarrow{j_1} & R_1 + R_2 & \xleftarrow{j_2} & R_2 \\ r_1 \downarrow & & r \downarrow & & \downarrow r_2 \\ \mathbf{F}R_1 & \xrightarrow{\mathbf{F}j_1} & \mathbf{F}(R_1 + R_2) & \xleftarrow{\mathbf{F}j_2} & \mathbf{F}R_2 \end{array}$$

as commuting diagram, where $j_i : R_i \rightarrow R_1 + R_2$ is the respective embedding, $i = 1, 2$.

2. We claim that the projections $\pi'_S : R_1 + R_2 \rightarrow S$ and $\pi'_T : R_1 + R_2 \rightarrow T$ are morphisms. We establish this property only for π'_S . First note that $\pi'_S \circ j_1 = \pi_S^{R_1}$, so that we have $f \circ \pi'_S \circ j_1 = \mathbf{F}(\pi'_S) \circ \mathbf{F}(j_1) \circ r_1 = \mathbf{F}(\pi'_S) \circ r \circ j_1$, similarly, $f \circ \pi'_S \circ j_2 = \mathbf{F}(\pi'_S) \circ r \circ j_2$. Thus we may conclude that $f \circ \pi'_S = \mathbf{F}(\pi'_S) \circ r$, so that indeed $\pi'_S : R_1 + R_2 \rightarrow S$ is a morphism.

3. Since $R_1 + R_2$ is a coalgebra, we know from Lemma 1.146 that $\langle \pi'_S, \pi'_T \rangle [R_1 + R_2]$ is a bisimulation. But this equals $R_1 \cup R_2$. \dashv

We briefly explore lattice properties for bisimulations on a coalgebra. For this, we investigate the union of an arbitrary family of bisimulations. Looking back at the union of two bisimulations, we used their sum as an intermediate construction. A more general consideration requires the sum of an arbitrary family. The following definition describes the coproduct as a specific form of a colimit, see Definition 1.90 .

Definition 1.156 *Let $(s_k)_{k \in I}$ be an arbitrary non-empty family of objects on a category \mathbf{K} . The object s together with morphisms $i_k : s_k \rightarrow s$ is called the coproduct of $(s_k)_{k \in I}$ iff given morphisms $j_k : s_k \rightarrow t$ for an object t there exists a unique morphism $j : s \rightarrow t$ with $j_k = j \circ i_k$ for all $k \in I$. s is denoted as $\sum_{k \in I} s_k$.*

Taking $I = \{1, 2\}$, one sees that the coproduct of two objects is in fact a special case of the coproduct just defined. The following diagram gives a general idea.

$$\begin{array}{ccccc} \dots & s_{r_1} & & \dots & s_{r_k} & \dots \\ & \searrow i_{r_1} & & \searrow i_{r_k} & & \\ & & s & & & \\ & \searrow j_{r_1} & \downarrow j & \swarrow j_{r_k} & & \\ & & t & & & \end{array}$$

The coproduct is uniquely determined up to isomorphisms.

Example 1.157 Consider the category **Set** of sets with maps as morphisms, and let $(S_k)_{k \in I}$ be a family of sets. Then

$$S := \bigcup_{k \in I} \{\langle s, k \rangle \mid s \in S_k\}$$

is a coproduct. In fact, $i_k : s \mapsto \langle s, k \rangle$ maps S_k to S , and if $j_k : S_k \rightarrow T$, put $j : S \rightarrow T$ with $j(s, k) := j_k(s)$, then $j_k = j \circ i_k$ for all k . \mathbb{M}

We put this new machinery to use right away, returning to our scenario given by functor \mathbf{F} .

Proposition 1.158 *Assume that \mathbf{F} preserves weak pullbacks. Let $(R_k)_{k \in I}$ be a family of bisimulations for coalgebras (S, f) and (T, g) . Then $\bigcup_{k \in I} R_k$ is a bisimulation for these coalgebras.*

Proof 1. Given $k \in I$, let $r_k : R_k \rightarrow \mathbf{F}R_k$ be the morphism on R_k such that $\pi_S : (S, f) \rightarrow (R_k, r_k)$ and $\pi_T : (T, g) \rightarrow (R_k, r_k)$ are morphisms for the coalgebras involved. Then there exists a unique coalgebra structure r on $\sum_{k \in I} R_k$ such that $i_\ell : (R_\ell, r_\ell) \rightarrow (\sum_{k \in I} R_k, r)$ is a coalgebra structure for all $\ell \in I$. This is shown exactly through the same argument as in the proof of Lemma 1.155 (*mutatis mutandis*: replace the coproduct of two bisimulations by the general coproduct).

2. The projections $\pi'_S : \sum_{k \in I} R_k \rightarrow S$ and $\pi'_T : \sum_{k \in I} R_k \rightarrow T$ are morphisms, and one shows exactly as in the proof of Lemma 1.155 that

$$\bigcup_{k \in I} R_k = \langle \pi'_S, \pi'_T \rangle \left[\sum_{k \in I} R_k \right].$$

An application of Lemma 1.146 now establishes the claim. \dashv

This may be applied to an investigation of the lattice structure on the set of all bisimulations between coalgebras.

Proposition 1.159 *Assume that \mathbf{F} preserves weak pullbacks. Let $(R_k)_{k \in I}$ be a non-empty family of bisimulations for coalgebras (S, f) and (T, g) . Then*

1. *There exists a smallest bisimulation R^* with $R_k \subseteq R^*$ for all k .*
2. *There exists a largest bisimulation R_* with $R_k \supseteq R_*$ for all k .*

Proof 1. We claim that $R^* = \bigcup_{k \in I} R_k$. It is clear that $R_k \subseteq R^*$ for all $k \in I$. If R' is a bisimulation on (S, f) and (T, g) with $R_k \subseteq R'$ for all k , then $\bigcup_k R_k \subseteq R'$, thus $R^* \subseteq R'$. In addition, R^* is a bisimulation by Proposition 1.158. This establishes part 1.

2. Put

$$\mathcal{R} := \{R \mid R \text{ is a bisimulation for } (S, f) \text{ and } (T, g) \text{ with } R \subseteq R_k \text{ for all } k\}$$

If $\mathcal{R} = \emptyset$, we put $R_* := \emptyset$, so we may assume that $\mathcal{R} \neq \emptyset$. Put $R_* := \bigcup \mathcal{R}$. By Proposition 1.158 this is a bisimulation for (S, f) and (T, g) with $R_k \subseteq R_*$ for all k . Assume that R' is a bisimulation for (S, f) and (T, g) with $R' \subseteq R_k$ for all k , then $R' \in \mathcal{R}$, hence $R' \subseteq R_*$, so R_* is the largest one. This settles part 2. \dashv

Looking a bit harder at bisimulations for (S, f) alone, we find that the largest bisimulation is actually an equivalence relation. But we have to make sure first that a largest bisimulation exists at all.

Proposition 1.160 *If functor \mathbf{F} preserves weak pullbacks, then there exists a largest bisimulation R^* on coalgebra (S, f) . R^* is an equivalence relation.*

Proof 1. Let

$$\mathcal{R} := \{R \mid R \text{ is a bisimulation on } (S, f)\}.$$

Then $\Delta_S \in \mathcal{R}$, hence $\mathcal{R} \neq \emptyset$. We know from Lemma 1.144 that $R \in \mathcal{R}$ entails $R^{-1} \in \mathcal{R}$, and from Proposition 1.158 we infer that $R^* := \bigcup \mathcal{R} \in \mathcal{R}$. Hence R^* is a bisimulation on (S, f) .

2. R^* is even an equivalence relation.

- Since $\Delta_S \in \mathcal{R}$, we know that $\Delta_S \subseteq R^*$, thus R^* is reflexive.
- Because $R^* \in \mathcal{R}$ we conclude that $(R^*)^{-1} \in \mathcal{R}$, thus $(R^*)^{-1} \subseteq R^*$. Hence R^* is symmetric.
- Since $R^* \in \mathcal{R}$, we conclude from Proposition 1.151 that $R^* \circ R^* \in \mathcal{R}$, hence $R^* \circ R^* \subseteq R^*$. This means that R^* is transitive.

⊥

This has an interesting consequence. Given a bisimulation equivalence on a coalgebra, we do not only find a larger one which contains it, but we can also find a morphism between the corresponding factor spaces. To be specific:

Corollary 1.161 *Assume that functor \mathbf{F} preserves weak pullbacks, and that α is a bisimulation equivalence on (S, f) , then there exists a unique morphism $\tau_\alpha : (S/\alpha, f_\alpha) \rightarrow (S/R^*, f_{R^*})$, where $f_\alpha : S/\alpha \rightarrow \mathbf{F}(S/\alpha)$ and $f_{R^*} : S/R^* \rightarrow \mathbf{F}(S/R^*)$ are the induced dynamics.*

Proof 0. The dynamics $f_\alpha : S/\alpha \rightarrow \mathbf{F}(S/\alpha)$ and $f_{R^*} : S/R^* \rightarrow \mathbf{F}(S/R^*)$ exist by the definition of a bisimulation.

1. Define

$$\tau([s]_\alpha) := [s]_{R^*}$$

for $s \in S$. This is well defined. In fact, if $s \alpha s'$ we conclude by the maximality of R^* that $s R^* s'$, so $[s]_\alpha = [s']_\alpha$ implies $[s]_{R^*} = [s']_{R^*}$.

2. We claim that τ_α is a morphism, hence that the right hand side of this diagram commutes; the left hand side of the diagram is just for nostalgia.

$$\begin{array}{ccccc} S & \xrightarrow{\eta_\alpha} & S/\alpha & \xrightarrow{\tau_\alpha} & S/R^* \\ f \downarrow & & f_\alpha \downarrow & & \downarrow f_{R^*} \\ \mathbf{F}S & \xrightarrow{\mathbf{F}\eta_\alpha} & \mathbf{F}(S/\alpha) & \xrightarrow{\mathbf{F}\tau_\alpha} & \mathbf{F}(S/R^*) \end{array}$$

Now $\tau_\alpha \circ \eta_\alpha = \eta_{R^*}$, and the outer diagram commutes. The left diagram commutes because $\eta_\alpha : (S, f) \rightarrow (S/\alpha, f_\alpha)$ is a morphism, moreover, η_α is a surjective map. Hence the claim follows from Lemma 1.32, so that τ_α is a morphism indeed.

3. If τ'_α is another morphism with these properties, then we have $\tau'_\alpha \circ \eta_\alpha = \eta_{R^*} = \tau_\alpha \circ \eta_\alpha$, and since η_α is surjective, it is an epi by Proposition 1.23, which implies $\tau_\alpha = \tau'_\alpha$. ⊥

This is all well, but where do we get bisimulation equivalences from? If we cannot find examples for them, the efforts just spent may run dry. Fortunately, we are provided with ample bisimulation equivalences through coalgebra morphisms, specifically through their kernel (for a definition see page 12). It will turn out that each of these equivalences can be generated so.

Proposition 1.162 *Assume that \mathbf{F} preserves weak pullbacks, and that $\varphi : (S, f) \rightarrow (T, g)$ is a coalgebra morphism. Then $\ker(\varphi)$ is a bisimulation equivalence on (S, f) . Conversely, if α is a bisimulation equivalence on (S, f) , then there exists a coalgebra (T, g) and a coalgebra morphism $\varphi : (S, f) \rightarrow (T, g)$ with $\alpha = \ker(\varphi)$.*

Proof 1. We know that $\ker(\varphi)$ is an equivalence relation; since $\ker(\varphi) = \text{Graph}(\varphi) \circ \text{Graph}(\varphi)^{-1}$, we conclude from Corollary 1.145 that $\ker(\varphi)$ is a bisimulation.

2. Let α be a bisimulation equivalence on (S, f) , then the factor map $\eta_\alpha : (S, f) \rightarrow (S/\alpha, f_\alpha)$ is a morphism by Lemma 1.152, and $\ker(\eta_\alpha) = \{\langle s, s' \rangle \mid [s]_\alpha = [s']_\alpha\} = \alpha$. \dashv

1.6.2 Congruences

Bisimulations compare two systems with each other, while a congruence permits to talk about elements in a coalgebra which behave similar. Let us have a look at Abelian groups. An equivalence relation α on an Abelian group G , which is written additively, is a congruence iff $g \alpha h$ and $g' \alpha h'$ together imply $(g + g') \alpha (h + h')$. This means that α is compatible with the group structure; an equivalent formulation says that there exists a group structure on G/α such that the factor map $\eta_\alpha : G \rightarrow G/\alpha$ is a group morphism. Thus the factor map is the harbinger of the good news.

Definition 1.163 *Let (S, f) be an \mathbf{F} -coalgebra for the endofunctor \mathbf{F} on the category **Set** of sets. An equivalence relation α on S is called an \mathbf{F} -congruence iff there exists a coalgebra structure f_α on S/α such that $\eta_\alpha : (S, f) \rightarrow (S/\alpha, f_\alpha)$ is a coalgebra morphism.*

Thus we want that this diagram

$$\begin{array}{ccc} S & \xrightarrow{\eta_\alpha} & S/\alpha \\ f \downarrow & & \downarrow f_\alpha \\ \mathbf{F}S & \xrightarrow{\mathbf{F}\eta_\alpha} & \mathbf{F}(S/\alpha) \end{array}$$

is commutative, so that we have

$$f_\alpha([s]_\alpha) = (\mathbf{F}\eta_\alpha)(f(s))$$

for each $s \in S$. A brief look at Lemma 1.152 shows that bisimulation equivalences are congruences, and we see from Proposition 1.162 that the kernels of coalgebra morphisms are congruences, provided the functor \mathbf{F} preserves weak pullbacks.

Hence congruences and bisimulations on a coalgebra are actually very closely related. They are, however, not the same, because we have

Proposition 1.164 *Let $\varphi : (S, f) \rightarrow (T, g)$ be a morphism for the \mathbf{F} -coalgebras (S, f) and (T, g) . Assume that $\ker(\mathbf{F}\varphi) \subseteq \ker(\mathbf{F}\eta_{\ker(\varphi)})$. Then $\ker(\varphi)$ is a congruence for (S, f) .*

Proof Define $f_{\ker(\varphi)}([s]_{\ker(\varphi)}) := \mathbf{F}(\eta_{\ker(\varphi)})(f(s))$ for $s \in S$. Then $f_{\ker(\varphi)} : S/\ker(\varphi) \rightarrow \mathbf{F}(S/\ker(\varphi))$ is well defined. In fact, assume that $[s]_{\ker(\varphi)} = [s']_{\ker(\varphi)}$, then $g(\varphi(s)) = g(\varphi(s'))$, so that $(\mathbf{F}\varphi)(f(s)) = (\mathbf{F}\varphi)(f(s'))$, consequently $\langle f(s), f(s') \rangle \in \ker(\mathbf{F}\varphi)$. By assumption,

$(\mathbf{F}\eta_{\ker(\varphi)})(f(s)) = (\mathbf{F}\eta_{\ker(\varphi)})(f(s'))$, so that $f_{\ker(\varphi)}([s]_{\ker(\varphi)}) = f_{\ker(\varphi)}([s']_{\ker(\varphi)})$. It is clear that η_α is a coalgebra morphism.

⊥

The next example leaves the category of sets and considers the category of measurable spaces, introduced in Example 1.11. The subprobability functor \mathbb{S} , introduced in Example 1.70, is an endofunctor on **Meas**, and we know that the coalgebras for this functor are just the subprobabilistic transition kernels $K : (S, \mathcal{A}) \rightarrow (S, \mathcal{A})$, see Example 1.130.

Fix measurable spaces (S, \mathcal{A}) and (T, \mathcal{B}) . A measurable map $f : (S, \mathcal{A}) \rightarrow (T, \mathcal{B})$ is called *final* iff \mathcal{B} is the largest σ -algebra on T which renders f measurable, so that $\mathcal{A} = \{f^{-1}[B] \mid B \in \mathcal{B}\}$. Thus we conclude from $f^{-1}[B] \in \mathcal{A}$ that $B \in \mathcal{B}$. Given an equivalence relation α on S , we can make the factor space S/α a measurable space by endowing it with the final σ -algebra \mathcal{A}/α with respect to η_α , compare Exercise 25.

This is the definition then of a congruence for coalgebras for the Giry functor.

Definition 1.165 *Let (S, \mathcal{A}, K) and (T, \mathcal{B}, L) be coalgebras for the subprobability functor, then $\varphi : (S, \mathcal{A}, K) \rightarrow (T, \mathcal{B}, L)$ is a coalgebra morphism iff $\varphi : (S, \mathcal{A}) \rightarrow (T, \mathcal{B})$ is a measurable map such that this diagram commutes.*

$$\begin{array}{ccc} S & \xrightarrow{\varphi} & T \\ K \downarrow & & \downarrow L \\ \mathbb{S}(S, \mathcal{A}) & \xrightarrow{\mathbb{S}\varphi} & \mathbb{S}(T, \mathcal{B}) \end{array}$$

Thus we have

$$L(\varphi(s))(B) = \mathbb{S}(\varphi)(K(s))(B) = K(s)(\varphi^{-1}[B])$$

for each $s \in S$ and for each measurable set $B \in \mathcal{B}$. We will investigate the kernel of a morphism now in order to obtain a result similar to the one reported in Proposition 1.164. The crucial property in that development has been the comparison of the kernel $\ker(\mathbf{F}\varphi)$ with $\ker(\mathbf{F}\eta_{\ker(\varphi)})$. We will investigate this property now.

Call a morphism φ *strong* iff φ is surjective and final. Now fix a strong morphism $\varphi : K \rightarrow L$. A measurable subset $A \in \mathcal{A}$ is called φ -*invariant* iff $a \in A$ and $\varphi(a) = \varphi(a')$ together imply $a' \in A$, so that $A \in \mathcal{A}$ is φ invariant iff A is the union of $\ker(\varphi)$ -equivalence classes.

We obtain:

Lemma 1.166 *Let $\Sigma_\varphi := \{A \in \mathcal{A} \mid A \text{ is } \varphi\text{-invariant}\}$. Then*

1. Σ_φ is a σ -algebra.
2. Σ_φ is isomorphic to $\{\varphi^{-1}[B] \mid B \in \mathcal{B}\}$ as a Boolean σ -algebra.

Proof 1. Clearly, both \emptyset and S are φ -invariant, and the complement of an invariant set is invariant again. Invariant sets are closed under countable unions. Hence Σ_φ is a σ -algebra.

2. Given $B \in \mathcal{B}$, it is clear that $\varphi^{-1}[B]$ is φ -invariant; since the latter is also a measurable subset of S , we conclude that $\{\varphi^{-1}[B] \mid B \in \mathcal{B}\} \subseteq \Sigma_\varphi$. Now let $A \in \Sigma_\varphi$, we claim that

$A = \varphi^{-1}[\varphi[A]]$. In fact, since $\varphi(a) \in \varphi[A]$ for $a \in A$, the inclusion $A \subseteq \varphi^{-1}[\varphi[A]]$ is trivial. Let $a \in \varphi^{-1}[\varphi[A]]$, so that there exists $a' \in A$ with $\varphi(a) = \varphi(a')$. Since A is φ -invariant, we conclude $a \in A$, establishing the other inclusion. Because φ is final and surjective, we infer from this representation that $\varphi[A] \in \mathcal{B}$, whenever $A \in \Sigma_\varphi$, and that $\varphi^{-1} : \mathcal{B} \rightarrow \Sigma_\varphi$ is surjective. Since φ is surjective, φ^{-1} is injective, hence this yields a bijection. The latter map is compatible with the operations of a Boolean σ -algebra, so it is an isomorphism. \dashv

This helps in establishing the crucial property for kernels.

Corollary 1.167 *Let $\varphi : K \rightarrow L$ be a strong morphism, then $\ker(\mathbb{S}\varphi) \subseteq \ker(\mathbb{S}\eta_{\ker(\varphi)})$.*

Proof Let $\langle \mu, \mu' \rangle \in \ker(\mathbb{S}\varphi)$, thus $(\mathbb{S}\varphi)(\mu)(B) = (\mathbb{S}\varphi)(\mu')(B)$ for all $B \in \mathcal{B}$. Now let $C \in \mathcal{A}/\ker(\varphi)$, then $\eta_{\ker(\varphi)}^{-1}[C] \in \Sigma_\varphi$, so that there exists by Lemma 1.166 some $B \in \mathcal{B}$ such that $\eta_{\ker(\varphi)}^{-1}[C] = \varphi^{-1}[B]$. Hence

$$\begin{aligned} (\mathbb{S}\eta_{\ker(\varphi)})(\mu)(C) &= \mu(\eta_{\ker(\varphi)}^{-1}[C]) \\ &= \mu(\varphi^{-1}[B]) \\ &= (\mathbb{S}\varphi)(\mu)(B) \\ &= (\mathbb{S}\varphi)(\mu')(B) \\ &= (\mathbb{S}\eta_{\ker(\varphi)})(\mu')(C), \end{aligned}$$

so that $\langle \mu, \mu' \rangle \in \ker(\mathbb{S}\eta_{\ker(\varphi)})$. \dashv

Now everything is in place to show that the kernel of a strong morphism is a congruence for the \mathbb{S} -coalgebra (S, \mathcal{A}, K) .

Proposition 1.168 *Let $\varphi : K \rightarrow L$ be a strong morphism for the \mathbb{S} -coalgebras (S, \mathcal{A}, K) and (T, \mathcal{B}, L) . Then $\ker(\varphi)$ is a congruence for (S, \mathcal{A}, K) .*

Proof 1. We want define the coalgebra $K_{\ker(\varphi)}$ on $(S/\ker(\varphi), \mathcal{A}/\ker(\varphi))$ upon setting

$$K_{\ker(\varphi)}([s]_{\ker(\varphi)})(C) := (\mathbb{S}\eta_{\ker(\varphi)})(K(s))(C) (= K(s)(\eta_{\ker(\varphi)}^{-1}[C]))$$

for $C \in \mathcal{A}/\ker(\varphi)$, but we have to be sure that this is well defined. In fact, let $[s]_{\ker(\varphi)} = [s']_{\ker(\varphi)}$, which means $\varphi(s) = \varphi(s')$, hence $L(\varphi(s)) = L(\varphi(s'))$, so that $(\mathbb{S}\varphi)K(s) = (\mathbb{S}\varphi)K(s')$, because $\varphi : K \rightarrow L$ is a morphism. But the latter equality implies $\langle K(s), K(s') \rangle \in \ker(\mathbb{S}\varphi) \subseteq \ker(\mathbb{S}\eta_{\ker(\varphi)})$, the inclusion holding by Corollary 1.167. Thus we conclude $(\mathbb{S}\eta_{\ker(\varphi)})(K(s)) = (\mathbb{S}\eta_{\ker(\varphi)})(K(s'))$, so that $K_{\ker(\varphi)}$ is well defined indeed.

2. It is immediate that $C \mapsto K_{\ker(\varphi)}([s]_{\ker(\varphi)})(C)$ is a subprobability on $\mathcal{A}/\ker(\varphi)$ for fixed $s \in S$, so it remains to show that $t \mapsto K_{\ker(\varphi)}(t)(C)$ is a measurable map on the factor space $(S/\ker(\varphi), \mathcal{A}/\ker(\varphi))$. Let $q \in [0, 1]$, and consider for $C \in \mathcal{A}/\ker(\varphi)$ the set $G := \{t \in S/\ker(\varphi) \mid K_{\ker(\varphi)}(t)(C) < q\}$. We have to show that $G \in \mathcal{A}/\ker(\varphi)$. Because $C \in \mathcal{A}/\ker(\varphi)$, we know that $A := \eta_{\ker(\varphi)}^{-1}[C] \in \Sigma_\varphi$, hence it is sufficient to show that the set $H := \{s \in S \mid K(s)(A) < q\} \in \Sigma_f$. Since K is the dynamics of a \mathbb{S} -coalgebra, we know that $H \in \mathcal{A}$, so it remains to show that H is φ -invariant. Because $A \in \Sigma_f$, we infer from Lemma 1.166 that

$A = \varphi^{-1}[B]$ for some $B \in \mathcal{B}$. Now take $s \in H$ and assume $\varphi(s) = \varphi(s')$. Thus

$$\begin{aligned} K(s')(A) &= K(s')(\varphi^{-1}[B]) \\ &= (\mathbb{S}\varphi)(K(s'))(B) \\ &= L(\varphi(s'))(B) \\ &= L(\varphi(s))(B) \\ &= K(s)(A) \\ &< q, \end{aligned}$$

so that $H \in \Sigma_\varphi$ indeed. Because $H = \eta_{\ker(\varphi)}^{-1}[G]$, it follows that $G \in [\mathcal{A}]_{\ker(\varphi)}$, and we are done. \dashv

1.7 Modal Logics

This section will discuss modal logics and have a closer look at the interface between models for this logics and coalgebras. Thus the topics of this section may be seen as an application and illustration of coalgebras.

We will define the language for the formulas of modal logics, first for the conventional logics which permits expressing sentences like “it is possible that formula φ holds” or “formula φ holds necessarily”, then for an extended version, allowing for modal operators that govern more than one formula. The interpretation through Kripke models is discussed, and it becomes clear at least elementary elements of the language of categories is helpful in investigating these logics. For completeness, we also give the construction for the canonical model, displaying the elegant construction through the Lindenbaum Lemma.

It shows that coalgebras can be used directly in the interpretation of modal logics. We demonstrate that a set of predicate liftings define a modal logics, discuss briefly expressivity for these modal logics, and display an interpretation of CTL*, one of the basic logics for model checking, through coalgebras.

We fix a set Φ of *propositional letters*.

Definition 1.169 *The basic modal language $\mathcal{L}(\Phi)$ over Φ is given by this grammar*

$$\varphi ::= \perp \mid p \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \Diamond\varphi$$

We introduce additional operators

$$\begin{aligned} \top &:= \neg\perp \\ \varphi_1 \vee \varphi_2 &:= \neg(\neg\varphi_1 \wedge \neg\varphi_2) \\ \varphi_1 \rightarrow \varphi_2 &:= \neg\varphi_1 \vee \varphi_2 \\ \Box\varphi &:= \neg\Diamond\neg\varphi. \end{aligned}$$

The constant \perp denotes falsehood, consequently, $\top = \neg\perp$ denotes truth, negation \neg and conjunction \wedge should not come as a surprise; informally, $\Diamond\varphi$ means that it is possible that formula φ holds, while $\Box\varphi$ expresses that φ holds necessarily. Syntactically, this looks like propositional logic, extended by the modal operators \Diamond and \Box .

Before we have a look at the semantics of modal logic, we indicate that this logic is syntactically sometimes a bit too restricted; after all, the modal operators operate only on one argument at a time.

The extension we want should offer modal operators with more arguments. For this, we introduce the notion of a *modal similarity type* $\tau = (\mathbf{O}, \rho)$, which is a set \mathbf{O} of operators, each operator $\Delta \in \mathbf{O}$ has an arity $\rho(\Delta) \in \mathbb{N}_0$. Note that $\rho(\Delta) = 0$ is not excluded; these modal constants will not play a distinguished rôle, however, they are sometimes nice to have.

Clearly, the set $\{\diamond\}$ together with $\rho(\diamond) = 1$ is an example for such a modal similarity type.

Definition 1.170 *Given a modal similarity type $\tau = (\mathbf{O}, \rho)$ and the set Φ of propositional letters, the extended modal language $\mathcal{L}(\tau, \Phi)$ is given by this grammar:*

$$\varphi ::= \perp \mid \mathbf{p} \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi \mid \Delta(\varphi_1, \dots, \varphi_k)$$

with $\mathbf{p} \in \Phi$ and $\Delta \in \mathbf{O}$ such that $\rho(\Delta) = k$.

We also introduce for the general case operators which negate on the negation of the arguments of a modal operator; they are called *nablas*, the nabla ∇ of Δ is defined through ($\Delta \in \mathbf{O}, \rho(\Delta) = k$)

$$\nabla(\varphi_1, \dots, \varphi_k) := \neg \Delta(\neg \varphi_1, \dots, \neg \varphi_k)$$

Hence \Box is the nabla of \diamond ; this is the reason why we did not mention \Box in the example above — it is dependent on \diamond in a systematic way.

It is time to have a look at some examples.

Example 1.171 Let $\mathbf{O} = \{\mathbf{F}, \mathbf{P}\}$ with $\rho(\mathbf{F}) = \rho(\mathbf{P}) = 1$; the operator \mathbf{F} looks into the future, and \mathbf{P} into the past. This may be useful, e.g., when you are traversing a tree and are visiting an inner node. The future may then look at all nodes in its subtree, the past at all nodes on a path from the root to this tree.

Then $\tau_{\text{Fut}} := (\mathbf{O}, \rho)$ is a modal similarity type. If φ is a formula in $\mathcal{L}(\tau_{\text{Fut}}, \Phi)$, formula $\mathbf{F}\varphi$ is true iff φ will hold in the future, and $\mathbf{P}\varphi$ is true iff φ did hold in the past. The nablas are defined as

$$\begin{aligned} \mathbf{G}\varphi &:= \neg \mathbf{F}\neg \varphi & (\varphi \text{ will always be the case}) \\ \mathbf{H}\varphi &:= \neg \mathbf{P}\neg \varphi & (\varphi \text{ has always been the case}). \end{aligned}$$

Look at some formulas:

$\mathbf{P}\varphi \rightarrow \mathbf{G}\mathbf{P}\varphi$: If something has happened, it will always have happened.

$\mathbf{F}\varphi \rightarrow \mathbf{F}\mathbf{F}\varphi$: If φ will be true in the future, then it will be true in the future that φ will be true.

$\mathbf{G}\mathbf{F}\varphi \rightarrow \mathbf{F}\mathbf{G}\varphi$: If φ will be true in the future, then it will at some point be always true.




The next example deals with a simple model for sequential programs.

Example 1.172 Take Ψ as a set of atomic programs (think of elements of Ψ as executable program components). The set of programs is defined through this grammar

$$t ::= \psi \mid t_1 \cup t_2 \mid t_1; t_2 \mid t^* \mid \varphi?$$

with $\psi \in \Psi$ and φ a formula of the underlying modal logic.

Here $t_1 \cup t_2$ denotes the nondeterministic choice between programs t_1 and t_2 , $t_1; t_2$ is the sequential execution of t_1 and t_2 in that order, and t^* is iteration of program t a finite number of times (including zero). The program $\varphi?$ tests whether or not formula φ holds; $\varphi?$ serves as a guard: $(\varphi?; t_1) \cup (\neg\varphi?; t_2)$ tests whether φ holds, if it does t_1 is executed, otherwise, t_2 is. So the informal meaning of $\langle t \rangle \varphi$ is that formula φ holds after program t is executed (we use here and later an expression like $\langle t \rangle \varphi$ rather than the functional notation or just juxtaposition).

So, formally we have the modal similarity type $\tau_{\text{PDL}} := (\mathbf{O}, \rho)$ with $\mathbf{O} := \{\langle t \rangle \mid t \text{ is a program}\}$. This logic is known as PDL — propositional dynamic logic. 

The next example deal with games and a syntax very similar to the one just explored for PDL.

Example 1.173 We introduce two players, Angel and Demon, playing against each other, taking turns. So Angel starts, then Demon makes the next move, then Angel replies, etc.

For modelling game logic, we assume that we have a set Γ of simple games; the syntax for games looks like this:


$$g ::= \gamma \mid g_1 \cup g_2 \mid g_1 \cap g_2 \mid g_1; g_2 \mid g^d \mid g^* \mid g^\times \mid \varphi?$$

with $\gamma \in \Gamma$ and φ a formula of the underlying logic. The informal interpretation of $g_1 \cup g_2$, $g_1; g_2$, g^* and $\varphi?$ are as in PDL (Example 1.172), but as actions of player Angel. The actions of player Demon are indicated by


$g_1 \cap g_2$: Demon chooses between games g_1 and g_2 ; this is called *demonic choice* (in contrast to *angelic choice* $g_1 \cup g_2$).

g^\times : Demon decides to play game g a finite number of times (including not at all).

g^d : Angel and Demon change places.

Again, we indicate through $\langle g \rangle \varphi$ that formula φ holds after game g . We obtain the similarity type $\tau_{\text{GL}} := (\mathbf{O}, \rho)$ with $\mathbf{O} := \{\langle g \rangle \mid g \text{ is a game}\}$ and $\rho = 1$. The corresponding logic is called *game logic* 

Another example is given by arrow logic. Assume that you have arrows in the plane; you can compose them, i.e., place the beginning of one arrow at the end of the first one, and you can reverse them. Finally, you can leave them alone, i.e., do nothing with an arrow.

Example 1.174 The set \mathbf{O} of operators for arrow logic is given by $\{\circ, \otimes, \text{skip}\}$ with $\rho(\circ) = 2$, $\rho(\otimes) = 1$ and $\rho(\text{skip}) = 0$. The arrow composed from arrows a_1 and a_2 is arrow $a_1 \circ a_2$, $\otimes a_1$ is the reversed arrow a_1 , and **skip** does nothing. 

1.7.1 Frames and Models

For interpreting the basic modal language, we introduce frames. A frame models transitions, which are at the very heart of modal logics. Let us have a brief look at a modal formula like $\Box p$ for some propositional letter $p \in \Phi$. This formula models “ p always holds”, which implies a transition from the current state to another one, in which p always holds; without a transition, we would not have to think whether p always holds — it would just hold or not. Hence we need to have transitions at our disposal, thus a transition system, as in Example 1.9. In the current context, we take the disguise of a transition system as a relation. All this is captured in the notion of a frame.

Definition 1.175 A Kripke frame $\mathfrak{F} := (W, R)$ for the basic modal language is a set $W \neq \emptyset$ of states together with a relation $R \subseteq W \times W$. W is sometimes called the set of worlds, R the accessibility relation.

The access-ability relation of a Kripke frame does not yet carry enough information about the meaning of a modal formula, since the propositional letters are not captured by the frame. This is the case, however, in a Kripke model.

Definition 1.176 A Kripke model (or simply a model) $\mathfrak{M} = (W, R, V)$ for the basic modal language consists of a Kripke frame (W, R) together with a map $V : \Phi \rightarrow \mathcal{P}(W)$.

So, roughly speaking, the frame part of a Kripke model caters for the propositional and the modal part of the logic whereas the map V takes care of the propositional letters. This now permits us to define the meaning of the formulas for the basic modal language. We state under which conditions a formula φ is true in a world $w \in W$; this is expressed through $\mathfrak{M}, w \models \varphi$; note that this will depend on the model \mathfrak{M} , hence we incorporate it usually into the notation. Here we go.

$\mathfrak{M}, w \models \perp$ is always false.

$\mathfrak{M}, w \models p \Leftrightarrow w \in V(p)$, if $p \in \Phi$.

$\mathfrak{M}, w \models \varphi_1 \wedge \varphi_2 \Leftrightarrow \mathfrak{M}, w \models \varphi_1$ and $\mathfrak{M}, w \models \varphi_2$.

$\mathfrak{M}, w \models \neg \varphi \Leftrightarrow \mathfrak{M}, w \models \varphi$ is false.

$\mathfrak{M}, w \models \Diamond \varphi \Leftrightarrow$ there exists v with $\langle w, v \rangle \in R$ and $\mathfrak{M}, v \models \varphi$.

The interesting part is of course the last line. We want $\Diamond \varphi$ to hold in state w ; by our informal understanding this means that a transition into a state such that φ holds in this state is possible. But this means that there exists some state v with $\langle w, v \rangle \in R$ such that φ holds in v . This is just the formulation we did use above. Look at $\Box \varphi$; an easy calculation shows that $\mathfrak{M}, w \models \Box \varphi$ iff $\mathfrak{M}, w \models \varphi$ for all v with $\langle w, v \rangle \in R$; thus, no matter what transition from world w to another world v we make, and $\mathfrak{M}, v \models \varphi$ holds, then $\mathfrak{M}, w \models \Box \varphi$.

We define $\llbracket \varphi \rrbracket_{\mathfrak{M}}$ as the set of all states in which formula φ holds. Formally,

$$\llbracket \varphi \rrbracket_{\mathfrak{M}} := \{w \in W \mid \mathfrak{M}, w \models \varphi\}.$$

Let us look at some examples.

Example 1.177 Put $\Phi := \{p, q, r\}$ as the set of propositional letters, $W := \{1, 2, 3, 4, 5\}$ as the set of states; relation R is given through

$$1 \longrightarrow 2 \longrightarrow 3 \longrightarrow 4 \longrightarrow 5$$

Finally, put

$$V(\ell) := \begin{cases} \{2, 3\}, & \ell = p \\ \{1, 2, 3, 4, 5\}, & \ell = q \\ \emptyset, & \ell = r \end{cases}$$

Then we have for the Kripke model $\mathfrak{M} := (W, R, V)$ for example

$\mathfrak{M}, 1 \models \Diamond \Box p$: This is so since $\mathfrak{M}, 3 \models p$ (because $3 \in V(p)$), thus $\mathfrak{M}, 2 \models \Box p$, hence $\mathfrak{M}, 1 \models \Diamond \Box p$.

$\mathfrak{M}, 1 \not\models \Diamond \Box p \rightarrow p$: Since $1 \notin V(p)$, we have $\mathfrak{M}, 1 \not\models p$.

$\mathfrak{M}, 2 \models \Diamond(p \wedge \neg r)$: The only successor to 2 in R is state 3, and we see that $3 \in V(p)$ and $3 \notin V(r)$.

$\mathfrak{M}, 1 \models q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond q)))$: Because $1 \in V(q)$ and 2 is the successor to 1, we investigate whether $\mathfrak{M}, 2 \models q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond q))$ holds. Since $2 \in V(q)$ and $\langle 2, 3 \rangle \in R$, we look at $\mathfrak{M}, 3 \models q \wedge \Diamond(q \wedge \Diamond q)$; now $\langle 3, 4 \rangle \in R$ and $\mathfrak{M}, 3 \models q$, so we investigate $\mathfrak{M}, 4 \models q \wedge \Diamond q$. Since $4 \in V(q)$ and $\langle 4, 5 \rangle \in R$, we find that this is true. Let φ denote the formula $q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond(q \wedge \Diamond q)))$, then this peeling off layers of parentheses shows that $\mathfrak{M}, 2 \not\models \varphi$, because $\mathfrak{M}, 5 \models \Diamond p$ does not hold.

$\mathfrak{M}, 1 \not\models \Diamond \varphi \wedge q$: Since $\mathfrak{M}, 2 \not\models \varphi$, and since state 2 is the only successor to 1, we see that $\mathfrak{M}, 1 \not\models \Diamond \varphi$.

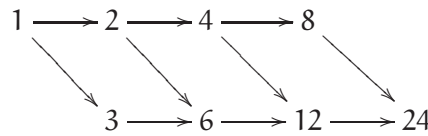
$\mathfrak{M}, w \models \Box q$: This is true for all worlds w , because $w' \in V(q)$ for all w' which are successors to some $w \in W$.



Example 1.178 We have two propositional letters p and q , as set of states we put $W := \{1, 2, 3, 4, 6, 8, 12, 24\}$, and we say

$$x R y \Leftrightarrow x \neq y \text{ and } x \text{ divides } y.$$

This is what R looks like without transitive arrows:



Put $V(p) := \{4, 8, 12, 24\}$ and $V(q) := \{6\}$. Define the Kripke model $\mathfrak{M} := (W, R, V)$, then we obtain for example

$\mathfrak{M}, 4 \models \Box p$: The set of successor to state 4 is just $\{8, 12, 24\}$ which is a subset of $V(p)$.

$\mathfrak{M}, 6 \models \Box p$: Here we may reason in the same way.

$\mathfrak{M}, 2 \not\models \Box p$: State 6 is a successor to 2, but $6 \notin V(p)$.

$\mathfrak{M}, 2 \models \Diamond(q \wedge \Box p) \wedge \Diamond(\neg q \wedge \Box p)$: State 6 is a successor to state 2 with $\mathfrak{M}, 6 \models q \wedge \Box p$, and state 4 is a successor to state 2 with $\mathfrak{M}, 4 \models \neg q \wedge \Box p$



Let us introduce some terminology which will be needed later. We say that a formula φ is *globally true* in a Kripke model \mathfrak{M} with state space W iff $\llbracket \varphi \rrbracket_{\mathfrak{M}} = W$, hence iff $\mathfrak{M}, w \models \varphi$ for all states $w \in W$; this is indicated by $\mathfrak{M} \models \varphi$. If $\llbracket \varphi \rrbracket_{\mathfrak{M}} \neq \emptyset$, thus if there exists $w \in W$ with $\mathfrak{M}, w \models \varphi$, we say that formula φ is *satisfiable*; φ is said to be *refutable* or *falsifiable* iff $\neg\varphi$ is satisfiable. A set Σ of formulas is said to be *globally true* iff $\mathfrak{M}, w \models \Sigma$ for all $w \in W$ (where we put $\mathfrak{M}, w \models \Sigma$ iff $\mathfrak{M}, w \models \varphi$ for all $\varphi \in \Sigma$). Σ is *satisfiable* iff $\mathfrak{M}, w \models \Sigma$ for some $w \in W$.

Kripke models are but one approach for interpreting modal logics. We observe that for a given transition system (S, \rightsquigarrow) the set $N(s) := \{s' \in S \mid s \rightsquigarrow s'\}$ may consist of more than one state; one may consider $N(s)$ as the neighborhood of state s . An external observer may not be able to observe $N(s)$ exactly, but may determine that $N(s) \subseteq A$ for some subset $A \subseteq S$. Obviously, $N(s) \subseteq A$ and $A \subseteq B$ implies $N(s) \subseteq B$, so that the sets defined by containing the neighborhood $N(s)$ of a state s forms an upper closed set. This leads to the definition of neighborhood frames.

Definition 1.179 Given a set S of states, a neighborhood frame $\mathfrak{N} := (S, N)$ is defined by a map $N : S \rightarrow \mathbf{V}(S) := \{V \subseteq \mathcal{P}(S) \mid V \text{ is upper closed}\}$.

The set $\mathbf{V}(S)$ of all upper closed families of subsets S was introduced Example 1.71.

So if we consider state $s \in S$ in a neighborhood frame, then $N(s)$ is an upper closed set which gives all sets the next state may be a member of. These frames occur in a natural way in topological spaces.

Example 1.180 Let (T, τ) be a topological space, then

$$V(t) := \{A \subseteq T \mid U \subseteq A \text{ for some open neighborhood } U \text{ of } t\}$$

defines a neighborhood frame (T, V) . 

Another straightforward example is given by ultrafilters.

Example 1.181 Given a set S , define

$$U(x) := \{U \subseteq S \mid x \in U\},$$

the ultrafilter associated with x . Then (S, U) is a neighborhood frame. 

Each Kripke frame gives rise to a neighborhood frame in this way:

Example 1.182 Let (W, R) be a Kripke frame, and define for the world $w \in W$ the set

$$V_R(w) := \{A \in \mathcal{P}(W) \mid R(w) \subseteq A\},$$

(with $R(w) := \{v \in W \mid \langle w, v \rangle \in R\}$), then plainly (W, V_R) is a neighborhood frame. 

A neighborhood frame induces a map on the power set of the state space into this power set. This map is used sometimes for an interpretation in lieu of the neighborhood function. Fix a map $P : S \rightarrow \mathbf{V}S$ for illustrating this. Given a subset $A \subseteq S$, we determine those states $\tau_P(A)$ which can achieve a state in A through P ; hence $\tau_P(A) := \{s \in S \mid A \in P(s)\}$. This yields a map $\tau_P : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$, which is monotone since $P(s)$ is upward closed for each s . Conversely, given a monotone map $\theta : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$, we define $P_\theta : S \rightarrow \mathbf{V}(S)$ through $R_\theta(s) := \{A \subseteq S \mid s \in \theta(A)\}$. It is plain that $\tau_{R_\theta} = \theta$ and $R_{\tau_P} = P$.

Definition 1.183 *Given a set S of states, a neighborhood frame (S, N) , and a map $V : \Phi \rightarrow \mathcal{P}(S)$, associating each propositional letter with a set of states. Then $\mathcal{N} := (S, N, V)$ is called a neighborhood model*

We define validity in a neighborhood model by induction on the structure of a formula, this time through the validity sets.

$$\begin{aligned} \llbracket \top \rrbracket_{\mathcal{N}} &:= S, \\ \llbracket p \rrbracket_{\mathcal{N}} &:= V(p), \text{ if } p \in \Phi, \\ \llbracket \varphi_1 \wedge \varphi_2 \rrbracket_{\mathcal{N}} &:= \llbracket \varphi_1 \rrbracket_{\mathcal{N}} \cap \llbracket \varphi_2 \rrbracket_{\mathcal{N}}, \\ \llbracket \neg \varphi \rrbracket_{\mathcal{N}} &:= S \setminus \llbracket \varphi \rrbracket_{\mathcal{N}}, \\ \llbracket \Box \varphi \rrbracket_{\mathcal{N}} &:= \{s \in S \mid \llbracket \varphi \rrbracket_{\mathcal{N}} \in N(s)\}. \end{aligned}$$

In addition, we put $\mathcal{N}, s \models \varphi$ iff $s \in \llbracket \varphi \rrbracket_{\mathcal{N}}$. Consider the last line and assume that the neighborhood frame underlying the model is generated by a Kripke frame (W, R) , so that $A \in N(w)$ iff $R(w) \subseteq A$. Then $\mathcal{N}, w' \models \Box \varphi$ translates into $w' \in \{w \in S \mid R(w) \subseteq \llbracket \varphi \rrbracket_{\mathcal{N}}\}$, so that $\mathcal{N}, w \models \Box \varphi$ iff each world which is accessible from world w satisfies φ ; this is what we want. Extending the definition above, we put

$$\llbracket \Diamond \varphi \rrbracket_{\mathcal{N}} := \{s \in S \mid S \setminus \llbracket \varphi \rrbracket_{\mathcal{N}} \notin N(s)\},$$

so that $\mathcal{N}, s \models \Diamond \varphi$ iff $\mathcal{N}, s \models \neg \Box \neg \varphi$.

We generalize the notion of a Kripke model for capturing extended modal languages. The idea for an extension is straightforward — for interpreting a modal formula given by a modal operator of arity n we require a subset of W^{n+1} . This leads to the definition of a frame, adapted to this purpose.

Definition 1.184 *Given a similarity type $\tau = (O, \rho)$, $\mathfrak{F} = (W, (R_\Delta)_{\Delta \in O})$ is said to be a τ -frame iff $W \neq \emptyset$ is a set of states, and $R_\Delta \subseteq W^{\rho(\Delta)+1}$ for each $\Delta \in O$. A τ -model $\mathfrak{M} = (\mathfrak{F}, V)$ is a τ -frame \mathfrak{F} with a map $V : \Phi \rightarrow \mathcal{P}(W)$.*

Given a τ -model \mathfrak{M} , we define the interpretation of formulas like $\Delta(\varphi_1, \dots, \varphi_n)$ and its nabla-cousin $\nabla(\varphi_1, \dots, \varphi_n)$ in this way:

- $\mathfrak{M}, w \models \Delta(\varphi_1, \dots, \varphi_n)$ iff there exist w_1, \dots, w_n with
 1. $\mathfrak{M}, w_i \models \varphi_i$ for $1 \leq i \leq n$,
 2. $\langle w, w_1, \dots, w_n \rangle \in R_\Delta$,
 if $n > 0$,
- $\mathfrak{M}, w \models \Delta$ iff $w \in R_\Delta$ for $n = 0$,


- $\mathfrak{M}, w \models \nabla(\varphi_1, \dots, \varphi_n)$ iff $(\langle w, w_1, \dots, w_n \rangle \in R_\Delta \text{ implies } \mathfrak{M}, w_i \models \varphi_i \text{ for all } i \in \{1, \dots, n\})$ for all $w_1, \dots, w_n \in W$, if $n > 0$,
- $\mathfrak{M}, w \models \nabla$ iff $w \notin R_\Delta$, if $n = 0$.

In the last two cases, ∇ is the nabla for modal operator Δ .

Just in order to get a grip on these definitions, let us have a look at some examples.

Example 1.185 The set O of modal operators consists just of the unary operators $\{\langle a \rangle, \langle b \rangle, \langle c \rangle\}$, the relations on the set $W := \{w_1, w_2, w_3, w_4\}$ of worlds are given by

$$\begin{aligned} R_a &:= \{\langle w_1, w_2 \rangle, \langle w_4, w_4 \rangle\}, \\ R_b &:= \{\langle w_2, w_3 \rangle\}, \\ R_c &:= \{\langle w_3, w_4 \rangle\}. \end{aligned}$$

There is only one propositional letter p and put $V(p) := \{w_2\}$. This comprises a τ -model \mathfrak{M} . We want to check whether $\mathfrak{M}, w_1 \models \langle a \rangle p \rightarrow \langle b \rangle p$ holds. Allora: In order to establish whether or not $\mathfrak{M}, w_1 \models \langle a \rangle p$ holds, we have to find a state v such that $\langle w_1, v \rangle \in R_a$ and $\mathfrak{M}, v \models p$; state w_2 is the only possible choice. But $\mathfrak{M}, w_1 \not\models p$, because $w_1 \notin V(p)$. Hence $\mathfrak{M}, w_1 \models \langle a \rangle p \rightarrow \langle b \rangle p$. 

Example 1.186 Let $W = \{u, v, w, s\}$ be the set of worlds, we take $O := \{\diamond, \clubsuit\}$ with $\rho(\diamond) = 2$ and $\rho(\clubsuit) = 3$. Put $R_\diamond := \{\langle u, v, w \rangle\}$ and $R_\clubsuit := \{\langle u, v, w, s \rangle\}$. The set Φ of propositional letters is $\{p_0, p_1, p_2\}$ with $V(p_0) := \{v\}$, $V(p_1) := \{w\}$ and $V(p_2) := \{s\}$. This yields a model \mathfrak{M} .

1. We want to determine $\llbracket \diamond(p_0, p_1) \rrbracket_{\mathfrak{M}}$. From the definition of \models we see that

$$\mathfrak{M}, x \models \diamond(p_0, p_1) \text{ iff } \exists x_0, x_1 : \mathfrak{M}, x_0 \models p_0 \text{ and } \mathfrak{M}, x_1 \models p_1 \text{ and } \langle x, x_0, x_1 \rangle \in R_\diamond.$$

We obtain by inspection $\llbracket \diamond(p_0, p_1) \rrbracket_{\mathfrak{M}} = \{u\}$.

2. We have $\mathfrak{M}, u \models \clubsuit(p_0, p_1, p_2)$. This is so since $\mathfrak{M}, v \models p_0$, $\mathfrak{M}, w \models p_1$, and $\mathfrak{M}, s \models p_2$ together with $\langle u, v, w, s \rangle \in R_\clubsuit$.
3. Consequently, we have $\llbracket \diamond(p_0, p_1) \rightarrow \clubsuit(p_0, p_1, p_2) \rrbracket_{\mathfrak{M}} = \{u\}$.



Example 1.187 Let's look into the future and into the past. We are given the unary operators $O = \{F, P\}$ as in Example 1.171. The interpretation requires two binary relations R_F and R_P ; we have defined the corresponding nablas G resp H . Unless we want to change the past, we assume that $R_P = R_F^{-1}$, so just one relation $R := R_F$ suffices for interpreting this logic. Hence

$\mathfrak{M}, x \models F\varphi$: This is the case iff there exists $z \in W$ such that $\langle x, z \rangle \in R$ and $\mathfrak{M}, z \models \varphi$.

$\mathfrak{M}, x \models P\varphi$: This is true iff there exists $z \in W$ with $\langle v, x \rangle \in R$ and $\mathfrak{M}, v \models \varphi$.

$\mathfrak{M}, x \models G\varphi$: This holds iff we have $\mathfrak{M}, y \models \varphi$ for all y with $\langle x, y \rangle \in R$.

$\mathfrak{M}, x \models H\varphi$: Similarly, for all y with $\langle y, x \rangle \in R$ we have $\mathfrak{M}, y \models \varphi$.



The next case is a little more complicated since we have to construct the relations from the information that is available. In the case of PDL (see Example 1.172), we have only information about the behavior of atomic programs, and we construct from it the relations for compound programs.

Example 1.188 Let Ψ be the set of all atomic programs, and assume that we have for each $t \in \Psi$ a relation $R_t \subseteq W \times W$; so if atomic program t is executed in state s , then $R_t(s)$ yields the set of all possible successor states after execution. Now we define by induction on the structure of the programs these relations.

$$\begin{aligned} R_{\pi_1 \cup \pi_2} &:= R_{\pi_1} \cup R_{\pi_2}, \\ R_{\pi_1; \pi_2} &:= R_{\pi_1} \circ R_{\pi_2}, \\ R_{\pi^*} &:= \bigcup_{n \geq 0} R_{\pi^n}. \end{aligned}$$

(here we put $R_{\pi_0} := \{\langle w, w \rangle \mid w \in W\}$, and $R_{\pi^{n+1}} := R_{\pi} \circ R_{\pi^n}$). Then, if $\langle x, y \rangle \in R_{\pi_1 \cup \pi_2}$, we know that $\langle x, y \rangle \in R_{\pi_1}$ or $\langle x, y \rangle \in R_{\pi_2}$, which reflects the observation that we can enter a new state y upon choosing between π_1 and π_2 . Hence executing $\pi_1 \cup \pi_2$ in state x , we should be able to enter this state upon executing one of the programs. Similarly, if in state z we execute first π_1 and then π_2 , we should enter an intermediate state z after executing π_1 and then execute π_2 in state z , yielding the resulting state. Executing π^* means that we execute π^n a finite number of times (probably not at all). This explains the definition for R_{π^*} .

Finally, we should define $R_{\varphi?}$ for a formula φ . The intuitive meaning of a program like $\varphi?; \pi$ is that we want to execute π , provided formula φ holds. This suggests defining

$$R_{\varphi?} := \{\langle w, w \rangle \mid \mathfrak{M}, w \models \varphi\}.$$

Note that we rely here on a model \mathfrak{M} which is already defined.

Just to get familiar with these definitions, let us have a look at the composition operator.

$$\begin{aligned} \mathfrak{M}, x \models \langle \pi_1; \pi_2 \rangle \varphi &\Leftrightarrow \exists v : \mathfrak{M}, v \models \varphi \text{ and } \langle x, v \rangle \in R_{\pi_1; \pi_2} \\ &\Leftrightarrow \exists w \in W \exists v \in \llbracket \varphi \rrbracket_{\mathfrak{M}} : \langle x, w \rangle \in R_{\pi_1} \text{ and } \langle w, v \rangle \in R_{\pi_2} \\ &\Leftrightarrow \exists w \in \llbracket \langle \pi_2 \rangle \varphi \rrbracket_{\mathfrak{M}} : \langle x, w \rangle \in R_{\pi_1} \\ &\Leftrightarrow \mathfrak{M}, x \models \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi \end{aligned}$$

This means that $\langle \pi_1; \pi_2 \rangle \varphi$ and $\langle \pi_1 \rangle \langle \pi_2 \rangle \varphi$ are semantically equivalent, which is intuitively quite clear.

The test operator is examined in the next formula. We have

$$R_{\varphi?; \pi} = R_{\varphi?} \circ R_{\pi} = \{\langle x, y \rangle \mid \mathfrak{M}, x \models \varphi \text{ and } \langle x, y \rangle \in R_{\pi}\} = R_{\varphi?} \cap R_{\pi}.$$

hence $\mathfrak{M}, y \models \langle \varphi?; \pi \rangle \psi$ iff $\mathfrak{M}, y \models \varphi$ and $\mathfrak{M}, y \models \langle \pi \rangle \psi$, so that

$$\mathfrak{M}, y \models (\langle \varphi?; \pi_1 \rangle \cup \langle \neg \varphi?; \pi_2 \rangle) \varphi \text{ iff } \begin{cases} \mathfrak{M}, y \models \langle \pi_1 \rangle \varphi, & \text{if } \mathfrak{M}, y \models \varphi \\ \mathfrak{M}, y \models \langle \pi_2 \rangle \varphi, & \text{otherwise} \end{cases}$$



The next example shows that we can interpret PDL in a neighborhood model as well.

Example 1.189 We associate with each atomic program $t \in \Psi$ of PDL an effectivity function E_t on the state space W . Hence if we execute t in state w , then $E_t(w)$ is the set of all subsets A of the states so that the next state is a member of A (we say that the program t can *achieve* a state in A). Hence $(W, (E_t)_{t \in \Psi})$ is a neighborhood frame. We have indicated that we can construct from a neighborhood function a relation (see page 92), so we put

$$R'_t(A) := \{w \in W \mid A \in E_t(w)\},$$

giving a monotone function $R'_t : \mathcal{P}(W) \rightarrow \mathcal{P}(W)$. This function can be extended to programs along the syntax for programs in the following way, which is very similar to the one for relations:

$$\begin{aligned} R'_{\pi_1 \cup \pi_2} &:= R'_{\pi_1} \cup R'_{\pi_2}, \\ R'_{\pi_1 ; \pi_2} &:= R'_{\pi_1} \circ R'_{\pi_2} \\ R'_{\pi^*} &:= \bigcup_{n \geq 0} R'_{\pi^n} \end{aligned}$$

with R'_{π^0} and R'_{π^n} defined as above.

Assume that we have again a function $V : \Phi \rightarrow \mathcal{P}(W)$, yielding a neighborhood model \mathcal{N} . The definition above are used now for the interpretation of formulas $\langle \pi \rangle \varphi$ through $\llbracket \langle \pi \rangle \varphi \rrbracket_{\mathcal{N}} := R'_{\pi}(\llbracket \varphi \rrbracket_{\mathcal{N}})$. The definition of $R_{\varphi?}$ carries over, so that this yields an interpretation of PDL. ✎

Turning to game logic (see Example 1.173), we note that neighborhood models are suited to interpret this logic as well. Assign for each atomic game $\gamma \in \Gamma$ to Angel the effectivity function P_{γ} , then $P_{\gamma}(s)$ indicates what Angel can achieve when playing γ in state s . Specifically, $A \in P_{\gamma}(s)$ indicates that Angel has a strategy for achieving that the next state of the game is a member of A by playing γ in state s . We will not formalize the notion of a strategy here but appeal rather to an informal understanding. The dual operator permit converting a game into its dual, where players change rôles: the moves of Angel become moves of Demon, and vice versa.

Let us just indicate informally by $\langle \gamma \rangle \varphi$ that Angel has a strategy in game γ which makes sure that game γ results in a state which satisfies formula φ . We assume the game to be *determined*: if one player does not have a winning strategy, then the other one has. Thus if Angel does not have a $\neg \varphi$ -strategy, then Demon has a φ -strategy, and vice versa.

Example 1.190 As in Example 1.173 we assume that games are given thorough this grammar

$$g ::= \gamma \mid g_1 \cup g_2 \mid g_1 \cap g_2 \mid g_1 ; g_2 \mid g^d \mid g^* \mid g^{\times} \mid \varphi?$$

with $\gamma \in \Gamma$, the set of atomic games. We assume that the game is determined, hence we may express demonic choice $g_1 \cap g_2$ through $(g_1^d \cup g_2^d)^d$, and demonic iteration g^{\times} through angelic iteration $((g^d)^*)^d$.

Assign to each $\gamma \in \Gamma$ an effectivity function P_{γ} on the set W of worlds, and put

$$P'_{\gamma}(A) := \{w \in W \mid A \in P_{\gamma}(w)\}.$$

Hence $w \in P'_\gamma(A)$ indicates that Angel has a strategy to achieve A by playing game γ in state w . We extend P' to games along the lines of the games' syntax:

$$\begin{aligned} P'_{g_1 \cup g_2}(A) &:= P'_{g_1}(A) \cup P'_{g_2}(A), & P'_{g^d}(A) &:= W \setminus P'_g(W \setminus A), \\ P'_{g_1; g_2}(A) &:= P'_{g_1}(P'_{g_2}(A)), & P'_{g_1 \cap g_2}(A) &:= P'_{(g_1^d \cup g_2^d)^d}(A), \\ P'_{g^*}(A) &:= \bigcup_{n \geq 0} P'_{g^n}(A), & P'_{g^\times}(A) &:= P'_{((g^d)^*)^d}(A), \\ P'_{\varphi?}(A) &:= \llbracket \varphi \rrbracket_{\mathcal{N}} \cap A. \end{aligned}$$

The last line refers to a model \mathcal{N} . \mathfrak{M}

We have finally a look at arrow logic, see Example 1.174.

Example 1.191 Arrows are interpreted as vectors, hence, e.g., as pairs. Let W be a set of states, then we take $W \times W$ as the domain of our interpretation. We have three modal operators.

- The nullary operator **skip** is interpreted through $R_{\text{skip}} := \{\langle w, w \rangle \mid w \in W\}$.
- The unary operator \otimes is interpreted through $R_{\otimes} := \{\langle \langle a, b \rangle, \langle b, a \rangle \rangle \mid a, b \in W\}$.
- The binary operator is intended to model composition, thus one end of the first arrow should be the be other end of the second arrow, hence $R_{\circ} := \{\langle \langle a, b \rangle, \langle b, c \rangle, \langle a, c \rangle \rangle \mid a, b, c \in W\}$.

With this, we obtain for example $\mathfrak{M}, \langle w_1, w_2 \rangle \models \psi_1 \circ \psi_2$ iff there exists v such that $\mathfrak{M}, \langle w_1, v \rangle \models \psi_1$ and $\mathfrak{M}, \langle v, w_2 \rangle \models \psi_2$. \mathfrak{M}

Frames are related through frame morphisms. Take a frame (W, R) for the basic modal language, then $R : W \rightarrow \mathcal{P}(W)$ is perceived as a coalgebra for the power set functor. this helps in defining morphisms.

Definition 1.192 Let $\mathfrak{F} = (W, R)$ and $\mathfrak{G} = (X, S)$ be Kripke frames. A frame morphism $f : \mathfrak{F} \rightarrow \mathfrak{G}$ is a map $f : W \rightarrow X$ which makes this diagram commutative:

$$\begin{array}{ccc} W & \xrightarrow{f} & X \\ R \downarrow & & \downarrow S \\ \mathcal{P}(W) & \xrightarrow{\mathcal{P}f} & \mathcal{P}(X) \end{array}$$

Hence we have for a frame morphism $f : \mathfrak{F} \rightarrow \mathfrak{G}$ the condition

$$S(f(w)) = (\mathcal{P}f)(R(w)) = f[R(w)] = \{f(w') \mid w' \in R(w)\}.$$

for all $w \in W$.

This is a characterization of frame morphisms.

Lemma 1.193 Let \mathfrak{F} and \mathfrak{G} be frames, as above. Then $f : \mathfrak{F} \rightarrow \mathfrak{G}$ is a frame morphism iff these conditions hold

1. $w R w'$ implies $f(w) S f(w')$.

2. If $f(w) S z$, then there exists $w' \in W$ with $z = f(w')$ and $w R w'$.

Proof 1. These conditions are necessary. In fact, if $\langle w, w' \rangle \in R$, then $f(w') \in f[R(w)] = S(f(w))$, so that $\langle f(w), f(w') \rangle \in S$. Similarly, assume that $f(w) S z$, thus $z \in S(f(w)) = \mathcal{P}(f)(R(w)) = f[R(w)]$. Hence there exists w' with $\langle w, w' \rangle \in R$ and $z = f(w')$.

2. The conditions are sufficient. The first condition implies $f[R(w)] \subseteq S(f(w))$. Now assume $z \in S(f(w))$, hence $f(w) S z$, thus there exists $w' \in R(w)$ with $f(w') = z$, consequently, $z = f(w') \in f[R(w)]$. \dashv

We see that the bounded morphisms from Example 1.10 appear here again in a natural context.

If we want to compare models for the basic modal language, then we certainly should be able to compare the underlying frames. But this is not yet enough, because the interpretation for atomic propositions has to be taken care of.

Definition 1.194 Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{N} = (X, S, Y)$ be models for the basic modal language and $f : (W, R) \rightarrow (X, S)$ be a frame morphism. Then $f : \mathfrak{M} \rightarrow \mathfrak{N}$ is said to be a model morphism iff $f^{-1} \circ Y = V$.

Hence $f^{-1}[Y(p)] = V(p)$ for a model morphism f and for each atomic proposition p , thus $\mathfrak{M}, w \models p$ iff $\mathfrak{N}, f(w) \models p$ for each atomic proposition. This extends to all formulas of the basic modal language.

Proposition 1.195 Assume \mathfrak{M} and \mathfrak{N} are models as above, and $f : \mathfrak{M} \rightarrow \mathfrak{N}$ is a model morphism. Then

$$\mathfrak{M}, w \models \varphi \text{ iff } \mathfrak{N}, f(w) \models \varphi$$

for all worlds w of \mathfrak{M} , and for all formulas φ .

Proof 0. The assertion is equivalent to

$$\llbracket \varphi \rrbracket_{\mathfrak{M}} = f^{-1}[\llbracket \varphi \rrbracket_{\mathfrak{N}}]$$

for all formulas φ . This is established by induction on the structure of a formula now.

1. If p is an atomic proposition, then this is just the definition of a frame morphism to be a model morphism:

$$\llbracket p \rrbracket_{\mathfrak{M}} = V(p) = f^{-1}[Y(p)] = \llbracket p \rrbracket_{\mathfrak{N}}.$$

Assume that the assertion holds for φ_1 and φ_2 , then

$$\begin{aligned} \llbracket \varphi_1 \wedge \varphi_2 \rrbracket_{\mathfrak{M}} &= \llbracket \varphi_1 \rrbracket_{\mathfrak{M}} \cap \llbracket \varphi_2 \rrbracket_{\mathfrak{M}} = f^{-1}[\llbracket \varphi_1 \rrbracket_{\mathfrak{N}}] \cap f^{-1}[\llbracket \varphi_2 \rrbracket_{\mathfrak{N}}] = \\ &= f^{-1}[\llbracket \varphi_1 \rrbracket_{\mathfrak{N}} \cap \llbracket \varphi_2 \rrbracket_{\mathfrak{N}}] = f^{-1}[\llbracket \varphi_1 \wedge \varphi_2 \rrbracket_{\mathfrak{N}}] \end{aligned}$$

Similarly, one shows that $\llbracket \neg \varphi \rrbracket_{\mathfrak{M}} = f^{-1}[\llbracket \neg \varphi \rrbracket_{\mathfrak{N}}]$.

2. Now consider $\Diamond\varphi$, assume that the hypothesis holds for formula φ , then we have

$$\begin{aligned}
 \llbracket \Diamond\varphi \rrbracket_{\mathfrak{M}} &= \{w \mid \exists w' \in R(w) : w' \in \llbracket \varphi \rrbracket_{\mathfrak{M}}\} \\
 &= \{w \mid \exists w' \in R(w) : f(w') \in \llbracket \varphi \rrbracket_{\mathfrak{N}}\} && \text{(by hypothesis)} \\
 &= \{w \mid \exists w' : f(w') \in S(f(w)), f(w') \in \llbracket \varphi \rrbracket_{\mathfrak{N}}\} && \text{(by Lemma 1.193)} \\
 &= f^{-1} [\{x \mid \exists x' \in S(x) : x' \in \llbracket \varphi \rrbracket_{\mathfrak{N}}\}] \\
 &= f^{-1} [\llbracket \Diamond\varphi \rrbracket_{\mathfrak{N}}]
 \end{aligned}$$

Thus the assertion holds for all formulas φ . \dashv

This permits comparing worlds in two models. Two worlds are said to be equivalent iff they cannot be separated by a formula, i.e., iff they satisfy exactly the same formulas.

Definition 1.196 *Let \mathfrak{M} and \mathfrak{N} be models with state spaces W resp. X . States $w \in W$ and $x \in X$ are called modally equivalent iff we have*

$$\mathfrak{M}, w \models \varphi \text{ iff } \mathfrak{N}, x \models \varphi$$

for all formulas φ

Hence if $f : \mathfrak{M} \rightarrow \mathfrak{N}$ is a model morphism, then w and $f(w)$ are modally equivalent for each world w of \mathfrak{M} . One might be tempted to compare models with respect to their transition behavior; after all, underlying a model is a transition system, a.k.a. a frame. This leads directly to this notion of bisimilarity for models — note that we have to take the atomic propositions into account.

Definition 1.197 *Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{N} = (X, S, Y)$ be models for the basic modal language, then a relation $B \subseteq W \times X$ is called a bisimulation iff*

1. *If $w B x$, then w and x satisfy the same propositional letters (“atomic harmony”).*
2. *If $w B x$ and $w R w'$, then there exists x' with $x S x'$ and $w' B x'$ (forth condition).*
3. *If $w B x$ and $x S x'$, then there exists w' with $w R w'$ and $w' B x'$ (back condition).*

States w and x are called bisimilar iff there exists a bisimulation B with $\langle w, x \rangle \in B$.

Hence the forth condition says for a pair of worlds $\langle w, x \rangle \in B$ that, if $w \rightsquigarrow_R w'$ there exists x' with $\langle w', x' \rangle \in B$ such that $x \rightsquigarrow_S x'$, similarly for the back condition. So this rings a bell: we did discuss this in Definition 1.138. Consequently, if models \mathfrak{M} and \mathfrak{N} are bisimilar, then the underlying frames are bisimilar coalgebras.

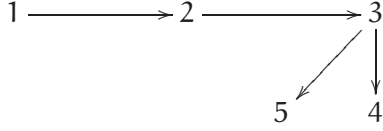
Consider this example for bisimilar states.

Example 1.198 Let relation B be defined through

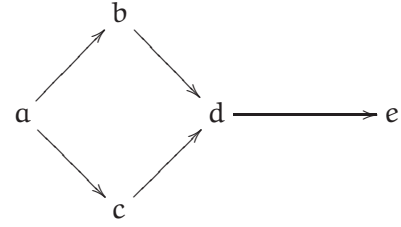
$$B := \{\langle 1, a \rangle, \langle 2, b \rangle, \langle 2, c \rangle, \langle 3, d \rangle, \langle 4, e \rangle, \langle 5, e \rangle\}$$

with $V(p) := \{a, d\}$, $V(q) := \{b, c, e\}$.

The transitions for \mathfrak{M} are given through



\mathfrak{N} is given through



Then B is a bisimulation \mathfrak{M}

The first result relating bisimulation and modal equivalence is intuitively quite clear. Since a bisimulation reflects the structural similarity of the transition structure of the underlying transition systems, and since the validity of modal formulas is determined through this transition structure (and the behavior of the atomic propositional formulas), it does not come as a surprise that bisimilar states are modal equivalent.

Proposition 1.199 *Let \mathfrak{M} and \mathfrak{N} be models with states w and x . If w and x are bisimilar, then they are modally equivalent.*

Proof 0. Let B be the bisimulation for which we know that $\langle w, x \rangle \in B$. We have to show that

$$\mathfrak{M}, w \models \varphi \Leftrightarrow \mathfrak{N}, x \models \varphi$$

for all formulas φ . This is done by induction on the formula.

1. Because of atomic harmony, the equivalence holds for propositional formulas. It is also clear that conjunction and negation are preserved under this equivalence, so that the remaining (and interesting) case of proving the equivalence for a formula $\Diamond\varphi$ under the assumption that it holds for φ .

“ \Rightarrow ” Assume that $\mathfrak{M}, w \models \Diamond\varphi$ holds. Thus there exists a world w' in \mathfrak{M} with $w R w'$ and $\mathfrak{M}, w' \models \varphi$. Hence there exists by the forward condition a world x' in \mathfrak{N} with $x S x'$ and $\langle w', x' \rangle \in B$ such that $\mathfrak{N}, x' \models \varphi$ by the induction hypothesis. Because x' is a successor to x , we conclude $\mathfrak{N}, x \models \Diamond\varphi$.

“ \Leftarrow ” This is shown in the same way, using the back condition for B . \dashv

The converse holds only under the restrictive condition that the models are image finite. Thus each state has only a finite number of successor states; formally, model (W, R, V) is called *image finite* iff for each world w the set $R(w)$ is finite. Then the famous Hennessy-Milner Theorem says

Theorem 1.200 *If the models \mathfrak{M} and \mathfrak{N} are image finite, then modal equivalent states are bisimilar.*

Proof 1. Given two modal equivalent states w^* and x^* , we have to find a bisimulation B with $\langle w^*, x^* \rangle \in B$. The only thing we know about the states is that they are modally equivalent, hence that they satisfy exactly the same formulas. This suggests to define

$$B := \{ \langle w', x' \rangle \mid w' \text{ and } x' \text{ are modally equivalent} \}$$

and to establish B as a bisimulation. Since by assumption $\langle w^*, x^* \rangle \in B$, this will then prove the claim.

2. If $\langle w, x \rangle \in B$, then both satisfy the same atomic propositions by the definition of modal equivalence. Now let $\langle w, x \rangle \in B$ and $w R w'$. Assume that we cannot find x' with $x S x'$ and $\langle w', x' \rangle \in B$. We know that $\mathfrak{M}, w \models \Diamond \top$, because this says that there exists a successor to w , viz., w' . Since w and x satisfy the same formulas, $\mathfrak{N}, x \models \Diamond \top$ follows, hence $S(x) \neq \emptyset$. Let $S(x) = \{x_1, \dots, x_k\}$. Then, since w and x_i are not modally equivalent, we can find for each $x_i \in S(x)$ a formula ψ_i such that $\mathfrak{M}, w' \models \psi_i$, but $\mathfrak{N}, x_i \not\models \psi_i$. Hence $\mathfrak{M}, w \models \Diamond(\psi_1 \wedge \dots \wedge \psi_k)$, but $\mathfrak{N}, w \not\models \Diamond(\psi_1 \wedge \dots \wedge \psi_k)$. This is a contradiction, so the assumption is false, and we can find x' with $x S x'$ and $\langle w', x' \rangle \in B$.

The other conditions for a bisimulation are shown in exactly the same way. \dashv

Neighborhood models can be compared through morphisms as well. Recall that the functor \mathbf{V} underlies a neighborhood frame, see Example 1.71.

Definition 1.201 Let $\mathcal{N} = (W, N, V)$ and $\mathcal{M} = (X, M, Y)$ be neighborhood models for the basic modal language. A map $f : W \rightarrow X$ is called a neighborhood morphism $f : \mathcal{N} \rightarrow \mathcal{M}$ iff

- $N \circ f = (Vf) \circ M$,
- $V = f^{-1} \circ Y$.

A neighborhood morphism is a morphism for the neighborhood frame (the definition of which is straightforward), respecting the validity of atomic propositions. In this way, the definition follows the pattern laid out for morphisms of Kripke models.

Expanding the definition, $f : \mathcal{N} \rightarrow \mathcal{M}$ is a neighborhood morphism iff $B \in N(f(w))$ iff $f^{-1}[B] \in M(w)$ for all $B \subseteq X$ and all worlds $w \in W$, and iff $V(p) = f^{-1}[Y(p)]$ for all atomic sentences $p \in \Phi$. Morphisms for neighborhood models preserve validity in the same way as morphisms for Kripke models do:

Proposition 1.202 Let $f : \mathcal{N} \rightarrow \mathcal{M}$ be a neighborhood morphism for the neighborhood models $\mathcal{N} = (W, N, V)$ and $\mathcal{M} = (X, M, Y)$. Then

$$\mathcal{N}, w \models \varphi \Leftrightarrow \mathcal{M}, f(w) \models \varphi$$

for all formulas φ and for all states $w \in W$.

Proof The proof proceeds by induction on the structure of formula φ . The induction starts with φ an atomic proposition. The assertion is true in this case because of atomic harmony, see the proof of Proposition 1.195. We pick only the interesting modal case for the induction step. Hence assume the assertion is established for formula φ , then

$$\begin{aligned} \mathcal{M}, f(w) \models \Box \varphi &\Leftrightarrow \llbracket \varphi \rrbracket_{\mathcal{M}} \in M(f(w)) && \text{(by definition)} \\ &\Leftrightarrow f^{-1}[\llbracket \varphi \rrbracket_{\mathcal{M}}] \in N(w) && \text{(f is a morphism)} \\ &\Leftrightarrow \llbracket \varphi \rrbracket_{\mathcal{N}} \in N(w) && \text{(by induction hypothesis)} \\ &\Leftrightarrow \mathcal{N}, w \models \Box \varphi \end{aligned}$$

\dashv

We will not pursue this observation further at this point but rather turn to the construction of a canonic model. When we will discuss coalgebraic logics, however, this striking structural similarity of models and their morphisms will be shown to be the instance of more general phenomenon.

Before proceeding, we introduce the notion of a *substitution*, which is a map $\sigma : \Phi \rightarrow \mathcal{L}(\tau, \Phi)$. We extend a substitution in a natural way to formulas. Define by induction on the structure of a formula

$$\begin{aligned} p^\sigma &:= \sigma(p), \text{ if } p \in \Phi, \\ (\neg \varphi)^\sigma &:= \neg(\varphi^\sigma), \\ (\varphi_1 \wedge \varphi_2)^\sigma &:= \varphi_1^\sigma \wedge \varphi_2^\sigma, \\ (\Delta(\varphi_1, \dots, \varphi_k))^\sigma &:= \Delta(\varphi_1^\sigma, \dots, \varphi_k^\sigma), \text{ if } \Delta \in \mathbf{O} \text{ with } \rho(\Delta) = k. \end{aligned}$$

1.7.2 The Lindenbaum Construction

We will show now how we obtain from a set of formulas a model which satisfies exactly these formulas. The scenario is the basic modal language, and it is clear that not every set of formulas is in a position to generate such a model.

Let Λ be a set of formulas, then we say that

- Λ is *closed under modus ponens* iff $\varphi \in \Lambda$ and $\varphi \rightarrow \psi$ together imply $\psi \in \Lambda$;
- Λ is *closed under uniform substitution* iff given $\varphi \in \Lambda$ we may conclude that $\varphi^\sigma \in \Lambda$ for all substitutions σ .

These two closure properties turn out to be crucial for the generation of a model from a set of formulas. Those sets which satisfy them will be called modal logics, to be precise:

Definition 1.203 *Let Λ be a set of formulas of the basic modal language. Λ is called a modal logic iff these conditions are satisfied:*

1. Λ contains all propositional tautologies.
2. Λ is closed under modus ponens and under uniform substitution.

If formula $\varphi \in \Lambda$, then φ is called a theorem of Λ ; we write this as $\vdash_\Lambda \varphi$.

Example 1.204 These are some instances of elementary properties for modal logics.

1. If Λ_i is a modal logic for each $i \in I \neq \emptyset$, then $\bigcap_{i \in I} \Lambda_i$ is a modal logic. This is fairly easy to check.
2. We say for a formula φ and a frame \mathfrak{F} over W as a set of states that φ *holds in this frame* (in symbols $\mathfrak{F} \models \varphi$) iff $\mathfrak{M}, w \models \varphi$ for each $w \in W$ and each model \mathfrak{M} which is based on \mathfrak{F} . Let \mathbb{S} be a class of frames, then

$$\Lambda_{\mathbb{S}} := \bigcap_{\mathfrak{F} \in \mathbb{S}} \{\varphi \mid \mathfrak{F} \models \varphi\}$$

is a modal logic. We abbreviate $\varphi \in \Lambda_{\mathbb{S}}$ by $\mathbb{S} \models \varphi$.

3. Define similarly $\mathfrak{M} \models \varphi$ for a model \mathfrak{M} iff $\mathfrak{M}, w \models \varphi$ for each world w of \mathfrak{M} . Then put for a class \mathbb{M} of models

$$\Lambda_{\mathbb{M}} := \bigcap_{\mathfrak{M} \in \mathbb{M}} \{\varphi \mid \mathfrak{M} \models \varphi\}.$$

Then there are sets \mathbb{M} for which $\Lambda_{\mathbb{M}}$ is not a modal language. In fact, take a model \mathfrak{M} with world W and two propositional letters p, q with $V(p) = W$ and $V(q) \neq W$, then $\mathfrak{M}, w \models p$ for all w , hence $\mathfrak{M} \models p$, but $\mathfrak{M} \not\models q$. On the other hand, $q = p^\sigma$ under the substitution $\sigma : p \mapsto q$. Hence $\Lambda_{\{\mathfrak{M}\}}$ is not closed under uniform substitution.



This formalizes the notion of deduction:

Definition 1.205 Let Λ be a logic, and $\Gamma \cup \{\varphi\}$ a set of modal formulas.

- φ is deducible in Λ from Γ iff either \vdash_{Λ} , or if there exist formulas $\psi_1, \dots, \psi_k \in \Gamma$ such that $\vdash_{\Lambda} (\psi_1 \wedge \dots \wedge \psi_k) \rightarrow \varphi$. We write this down as $\Gamma \vdash_{\Lambda} \varphi$.
- Γ is Λ -consistent iff $\Gamma \not\vdash_{\Lambda} \perp$, otherwise Γ is called Λ -inconsistent.
- φ is called Λ -consistent iff $\{\varphi\}$ is Λ -consistent.

This is a simple and intuitive criterion for inconsistency. We fix for the discussions below a modal logic Λ .

Lemma 1.206 Let Γ be a set of formulas. Then these statements are equivalent

1. Γ is Λ -inconsistent.
2. $\Gamma \vdash_{\Lambda} \varphi \wedge \neg\varphi$ for some formula φ .
3. $\Gamma \vdash_{\Lambda} \psi$ for all formulas ψ .

Proof 1 \Rightarrow 2: Because $\Gamma \vdash_{\Lambda} \perp$, we know that $\psi_1 \wedge \dots \wedge \psi_k \rightarrow \perp$ is in Λ for some formulas $\psi_1, \dots, \psi_k \in \Gamma$. But $\perp \rightarrow \varphi \wedge \neg\varphi$ is a tautology, hence $\Gamma \vdash_{\Lambda} \varphi \wedge \neg\varphi$.

2 \Rightarrow 3: By assumption there exists $\psi_1, \dots, \psi_k \in \Gamma$ such that $\vdash_{\Lambda} \psi_1 \wedge \dots \wedge \psi_k \rightarrow \varphi \wedge \neg\varphi$, and $\varphi \wedge \neg\varphi \rightarrow \psi$ is a tautology for an arbitrary formula ψ , hence $\vdash_{\Lambda} \varphi \wedge \neg\varphi \rightarrow \psi$. Thus $\Gamma \vdash_{\Lambda} \psi$.

3 \Rightarrow 1: We have in particular $\Gamma \vdash_{\Lambda} \perp$. \dashv

Λ -consistent sets have an interesting compactness property.

Lemma 1.207 A set Γ of formulas is Λ -consistent iff each finite subset of Γ is Λ -consistent.

Proof If Γ is Λ -consistent, then certainly each finite subset is. If, on the other hand, each finite subset is Λ -consistent, then the whole set must be consistent, since consistency is tested with finite witness sets. \dashv

Proceeding on our path to finding a model for a modal logic, we define normal logics. They are closed under some properties which appear as fairly, well, normal, so it is not surprising that these normal logics will play an important rôle.

Definition 1.208 Modal logic Λ is called normal iff it satisfies these conditions for all propositional letters $p, q \in \Phi$ and all formulas φ :

(**K**) $\vdash_{\Lambda} \Box(p \rightarrow q) \rightarrow (\Box p \rightarrow \Box q)$,

(**D**) $\vdash_{\Lambda} \Diamond p \leftrightarrow \neg \Box \neg p$,

(**G**) *If $\vdash_{\Lambda} \varphi$, then $\vdash_{\Lambda} \Box \varphi$.*

Property (**K**) states that if it is necessary that p implies q , then the fact that p is necessary will imply that q is necessary. Note that the formulas in Λ do not have a semantics yet, they are for the time being just syntactic entities. Property (**D**) connects the constructors \Diamond and \Box in the desired manner. Finally, (**G**) states that, loosely speaking, if something is the case, then it is necessarily the case. We should finally note that (**K**) and (**D**) are both formulated for propositional letters only. This, however, is sufficient for modal logics, since they are closed under uniform substitution.

In a normal logic, the equivalence of formulas is preserved by the diamond.

Lemma 1.209 *Let Λ be a normal modal logic, then $\vdash_{\Lambda} \varphi \leftrightarrow \psi$ implies $\vdash_{\Lambda} \Diamond \varphi \leftrightarrow \Diamond \psi$.*

Proof We show that $\vdash_{\Lambda} \varphi \rightarrow \psi$ implies $\vdash_{\Lambda} \Diamond \varphi \rightarrow \Diamond \psi$, the rest will follow in the same way.

$$\begin{array}{ll}
 \vdash_{\Lambda} \varphi \rightarrow \psi \Rightarrow \vdash_{\Lambda} \neg \psi \rightarrow \neg \varphi & \text{(contraposition)} \\
 \Rightarrow \vdash_{\Lambda} \Box(\neg \psi \rightarrow \neg \varphi) & \text{(by (G))} \\
 \Rightarrow \vdash_{\Lambda} (\Box(\neg \psi \rightarrow \neg \varphi)) \rightarrow (\Box \neg \psi \rightarrow \Box \neg \varphi) & \text{(uniform substitution, (K))} \\
 \Rightarrow \vdash_{\Lambda} \Box \neg \psi \rightarrow \Box \neg \varphi & \text{(modus ponens)} \\
 \Rightarrow \vdash_{\Lambda} \neg \Box \neg \varphi \rightarrow \neg \Box \neg \psi & \text{(contraposition)} \\
 \Rightarrow \vdash_{\Lambda} \Diamond \varphi \rightarrow \Diamond \psi & \text{(by (D))}
 \end{array}$$

⊢

Let us define a semantic counterpart to $\Gamma \vdash_{\Lambda}$. Let \mathfrak{F} be a frame and Γ be a set of formulas, then we say that Γ holds on \mathfrak{F} (written as $\mathfrak{F} \models \Gamma$) iff each formula in Γ holds in each model which is based on frame \mathfrak{F} (see Example 1.204). We say that Γ entails formula φ ($\Gamma \models_{\mathfrak{F}} \varphi$) iff $\mathfrak{F} \models \Gamma$ implies $\mathfrak{F} \models \varphi$. This carries over to classes of frames in an obvious way. Let \mathbb{S} be a class of frames, then $\Gamma \models_{\mathbb{S}} \varphi$ iff we have $\Gamma \models_{\mathfrak{F}} \varphi$ for all frames $\mathfrak{F} \in \mathbb{S}$.

Definition 1.210 *Let \mathbb{S} be a class of frames, then the normal logic Λ is called \mathbb{S} -sound iff $\Lambda \subseteq \Lambda_{\mathbb{S}}$. If Λ is \mathbb{S} -sound, then \mathbb{S} is called a class of frames for Λ .*

Note that \mathbb{S} -soundness indicates that $\vdash_{\Lambda} \varphi$ implies $\mathfrak{F} \models \varphi$ for all frames $\mathfrak{F} \in \mathbb{S}$ and for all formulas φ .

This example dwells on traditional names.

Example 1.211 Let Λ_4 be the smallest modal logic which contains $\Diamond \Diamond p \rightarrow \Diamond p$ (if it is possible that p is possible, then p is possible), and let $\mathbf{K4}$ be the class of transitive frames. Then Λ_4 is $\mathbf{K4}$ -sound. In fact, it is easy to see that $\mathfrak{M}, w \models \Diamond \Diamond p \rightarrow \Diamond p$ for all worlds w , whenever \mathfrak{M} is a model the frame of which carries a transitive relation. \mathfrak{M}

Thus \mathbb{S} -soundness permits us to conclude that a formula which is deducible from Γ holds also in all frames from \mathbb{S} . Completeness goes the other way: roughly, if we know that a formula holds in a class of frames, then it is deducible. To be more precise:

Definition 1.212 Let \mathbb{S} be a class of frames and Λ a normal modal logic.

1. Λ is strongly \mathbb{S} -complete iff for any set $\Gamma \cup \{\varphi\}$ of formulas $\Gamma \models_{\mathbb{S}} \varphi$ implies $\Gamma \vdash_{\Lambda} \varphi$.
2. Λ is weakly \mathbb{S} -complete iff $\mathbb{S} \models \varphi$ implies $\vdash_{\Lambda} \varphi$ for any formula φ .

This is a characterization of completeness.

Proposition 1.213 Let Λ and \mathbb{S} be as above.

1. Λ is strongly \mathbb{S} -complete iff every Λ -consistent set of formulas is satisfiable for some $\mathfrak{F} \in \mathbb{S}$.
2. Λ is weakly \mathbb{S} -complete iff every Λ -consistent formula is satisfiable for some $\mathfrak{F} \in \mathbb{S}$.

Proof 1. If Λ is not strongly \mathbb{S} -complete, then we can find a set Γ of formulas and a formula φ with $\Gamma \models_{\mathbb{S}} \varphi$, but $\Gamma \not\vdash_{\Lambda} \varphi$. Then $\Gamma \cup \{\neg\varphi\}$ is Λ -consistent, but this set cannot be satisfied on \mathbb{S} . So the condition for strong completeness is sufficient. It is also necessary. In fact, we may assume by compactness that Γ is finite. Thus by consistency $\Gamma \not\vdash_{\Lambda} \perp$, hence $\Gamma \not\models_{\mathbb{S}} \perp$ by completeness, thus there exists a frame $\mathfrak{F} \in \mathbb{S}$ with $\mathfrak{F} \models \Gamma$ but $\mathfrak{F} \not\models \perp$.

2. This is but a special case of cardinality 1. \dashv

Consistent sets are not yet sufficient for the construction of a model, as we will see soon. We need consistent sets which cannot be extended further without jeopardizing their consistency. To be specific:

Definition 1.214 The set Γ of formulas is maximal Λ -consistent iff Γ is Λ -consistent, and it is not properly contained in a Λ -consistent set.

Thus if we have a maximal Λ -consistent set Γ , and if we know that $\Gamma \subset \Gamma_0$ with $\Gamma \neq \Gamma_0$, then we know that Γ_0 is not Λ -consistent. This criterion is sometimes a bit unpractical, but we have

Lemma 1.215 Let Λ be a normal logic and Γ be a maximally Λ -consistent set of formulas. Then

1. Γ is closed under modus ponens.
2. $\Lambda \subseteq \Gamma$.
3. $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$ for all formulas φ .
4. $\varphi \vee \psi \in \Gamma$ iff $\varphi \in \Gamma$ or $\psi \in \Gamma$ for all formulas φ, ψ .
5. $\varphi_1 \wedge \varphi_2 \in \Gamma$ if $\varphi_1, \varphi_2 \in \Gamma$.

Proof 1. Assume that $\varphi \in \Gamma$ and $\varphi \rightarrow \psi \in \Gamma$, but $\psi \notin \Gamma$. Then $\Gamma \cup \{\psi\}$ is inconsistent, hence $\Gamma \cup \{\psi\} \vdash_{\Lambda} \perp$ by Lemma 1.206. Thus we can find formulas $\psi_1, \dots, \psi_k \in \Gamma$ such that $\vdash_{\Lambda} \psi \wedge \psi_1 \wedge \dots \wedge \psi_k \rightarrow \perp$. Because $\vdash_{\Lambda} \varphi \wedge \psi_1 \wedge \dots \wedge \psi_k \rightarrow \psi \wedge \psi_1 \wedge \dots \wedge \psi_k$, we conclude $\Gamma \vdash_{\Lambda} \perp$. This contradicts Λ -consistency by Lemma 1.206.

2. In order to show that $\Lambda \subseteq \Gamma$, we assume that there exists $\psi \in \Lambda$ such that $\psi \notin \Gamma$, then $\Gamma \cup \{\psi\}$ is inconsistent, hence $\vdash_{\Lambda} \psi_1 \wedge \dots \wedge \psi_k \rightarrow \neg\psi$ for some $\psi_1, \dots, \psi_k \in \Lambda$ (here we use $\Gamma \cup \{\psi\} \vdash_{\Lambda} \psi$ and Lemma 1.206). By propositional logic, $\vdash_{\Lambda} \psi \rightarrow \neg(\psi_1 \wedge \dots \wedge \psi_k)$,

hence $\psi \in \Lambda$ implies $\Gamma \vdash_{\Lambda} \neg(\psi_1 \wedge \cdots \wedge \psi_k)$. But $\Gamma \vdash_{\Lambda} \psi_1 \wedge \cdots \wedge \psi_k$, consequently, Γ is Λ -inconsistent.

3. If both $\varphi \notin \Gamma$ and $\neg\varphi \notin \Gamma$, Γ is Λ -inconsistent.

4. Assume first that $\varphi \vee \psi \in \Gamma$, but $\varphi \notin \Gamma$ and $\psi \notin \Gamma$, hence both $\Gamma \cup \{\varphi\}$ and $\Gamma \cup \{\psi\}$ are inconsistent. Thus we can find $\psi_1, \dots, \psi_k, \varphi_1, \dots, \varphi_n \in \Gamma$ with $\vdash_{\Lambda} \psi_1 \wedge \cdots \wedge \psi_k \rightarrow \neg\psi$ and $\vdash_{\Lambda} \varphi_1 \wedge \cdots \wedge \varphi_n \rightarrow \neg\varphi$. This implies $\vdash_{\Lambda} \psi_1 \wedge \cdots \wedge \psi_k \wedge \varphi_1 \wedge \cdots \wedge \varphi_n \rightarrow \neg\psi \wedge \neg\varphi$, and by arguing propositionally, $\vdash_{\Lambda} (\psi \vee \varphi) \wedge \psi_1 \wedge \cdots \wedge \psi_k \wedge \varphi_1 \wedge \cdots \wedge \varphi_n \rightarrow \perp$, which contradicts Λ -consistency of Γ . For the converse, assume that $\varphi \in \Gamma$. Since $\varphi \rightarrow \varphi \vee \psi$ is a tautology, we obtain $\varphi \vee \psi$ from modus ponens.

5. Assume $\varphi_1 \wedge \varphi_2 \notin \Gamma$, then $\neg\varphi_1 \vee \neg\varphi_2 \in \Gamma$ by part 3. Thus $\neg\varphi_1 \in \Gamma$ or $\neg\varphi_2 \in \Gamma$ by part 4, hence $\varphi_1 \notin \Gamma$ or $\varphi_2 \notin \Gamma$. \dashv

Hence consistent sets have somewhat convenient properties, but how do we construct them? The famous Lindenbaum Lemma states that we may obtain them by enlarging consistent sets.

From now on we fix a normal modal logic Λ .

Lemma 1.216 *If Γ is a Λ -consistent set, then there exists a maximal Λ -consistent set Γ^+ with $\Gamma \subseteq \Gamma^+$.*

We will give two proofs for the Lindenbaum Lemma, depending on the cardinality of the set of all formulas. If the set Φ of propositional letters is countable, the set of all formulas is countable as well, so the first proof may be applied. If, however, we have more than a countable number of formulas, then this proof will fail to exhaust all formulas, and we have to apply another method, in this case transfinite induction (in the disguise of Tuckey's Lemma).

Proof (First — countable case) Assume that the set of all formulas is countable, and let $\{\varphi_n \mid n \in \mathbb{N}\}$ be an enumeration of them. Define by induction

$$\begin{aligned}\Gamma_0 &:= \Gamma, \\ \Gamma_{n+1} &:= \Gamma_n \cup \{\psi_n\},\end{aligned}$$

where

$$\psi_n := \begin{cases} \varphi_n, & \text{if } \Gamma_n \cup \{\varphi_n\} \text{ is consistent,} \\ \neg\varphi_n, & \text{otherwise.} \end{cases}$$

Put

$$\Gamma^+ := \bigcup_{n \in \mathbb{N}} \Gamma_n.$$

Then these properties are easily checked:

- Γ_n is consistent for all $n \in \mathbb{N}_0$.
- Either $\varphi \in \Gamma^+$ or $\neg\varphi \in \Gamma^+$ for all formulas φ .
- If $\Gamma^+ \vdash_{\Lambda} \varphi$, then $\varphi \in \Gamma^+$.
- Γ^+ is maximal.

⊥

Proof (Second — general case) Let

$$\mathbb{C} := \{\Gamma' \mid \Gamma' \text{ is } \Lambda\text{-consistent and } \Gamma \subseteq \Gamma'\}.$$

Then \mathbb{C} contains Γ , hence $\mathbb{C} \neq \emptyset$, and \mathbb{C} is ordered by inclusion. By Tuckey's Lemma, it contains a maximal chain \mathbb{C}_0 . Let $\Gamma^+ := \bigcup \mathbb{C}_0$. Then Γ^+ is a Λ -consistent set which contains Γ as a subset. While the latter is evident, we have to take care of the former. Assume that Γ^+ is not Λ -consistent, hence $\Gamma^+ \vdash_{\Lambda} \varphi \wedge \neg\varphi$ for some formula φ . Thus we can find $\psi_1, \dots, \psi_k \in \Gamma^+$ with $\vdash_{\Lambda} \psi_1 \wedge \dots \wedge \psi_k \rightarrow \varphi \wedge \neg\varphi$. Given $\psi_i \in \Gamma^+$, we can find $\Gamma_i \in \mathbb{C}_0$ with $\psi_i \in \Gamma_i$. Since \mathbb{C}_0 is linearly ordered, we find some Γ' among them such that $\Gamma_i \subseteq \Gamma'$ for all i . Hence $\psi_1, \dots, \psi_k \in \Gamma'$, so that Γ' is not Λ -consistent. This is a contradiction. Now assume that Γ^+ is not maximal, then there exists φ such that $\varphi \notin \Gamma^+$ and $\neg\varphi \notin \Gamma^+$. If $\Gamma^+ \cup \{\varphi\}$ is not consistent, $\Gamma^+ \cup \{\neg\varphi\}$ is, and vice versa, so either one of $\Gamma^+ \cup \{\varphi\}$ and $\Gamma^+ \cup \{\neg\varphi\}$ is consistent. But this means that \mathbb{C}_0 is not maximal. ⊥

We are in a position to construct a model now, specifically, we will define a set of states, a transition relation and the validity sets for the propositional letters. Put

$$W^{\#} := \{\Sigma \mid \Sigma \text{ is } \Lambda\text{-consistent and maximal}\},$$

$$R^{\#} := \{\langle w, v \rangle \in W^{\#} \times W^{\#} \mid \text{for all formulas } \psi, \psi \in v \text{ implies } \Diamond\psi \in w\},$$

$$V^{\#}(p) := \{w \in W^{\#} \mid p \in w\} \text{ for } p \in \Phi.$$

Then $\mathfrak{M}^{\#} := (W^{\#}, R^{\#}, V^{\#})$ is called the *canonical model* for Λ .

This is another view of relation $R^{\#}$:

Lemma 1.217 *Let $v, w \in W^{\#}$, then $wR^{\#}v$ iff $\Box\psi \in w$ implies $\psi \in v$ for all formulas ψ .*

Proof 1. Assume that $\langle w, v \rangle \in R^{\#}$, but that $\psi \notin v$ for some formula ψ . Since v is maximal, we conclude from Lemma 1.215 that $\neg\psi \in v$, hence the definition of $R^{\#}$ tells us that $\Diamond\neg\psi \in w$, which in turn implies by the maximality of w that $\neg\Diamond\neg\psi \notin w$, hence $\Box\psi \notin w$.

2. If $\Diamond\psi \notin w$, then by maximality $\neg\Diamond\psi \in w$, so $\Box\neg\psi \in w$, which means by assumption that $\neg\psi \in v$. Hence $\psi \notin v$. ⊥

The next lemma gives a more detailed look at the transitions which are modelled by $R^{\#}$.

Lemma 1.218 *Let $w \in W^{\#}$ with $\Diamond\varphi \in w$. Then there exists a state $v \in W^{\#}$ such that $\varphi \in v$ and $wR^{\#}v$.*

Proof Because we can extend Λ -consistent sets to maximal consistent ones by the Lindenbaum Lemma 1.216, it is enough to show that $v_0 := \{\varphi\} \cup \{\psi \mid \Box\psi \in w\}$ is Λ -consistent. Assume it is not. Then we have $\vdash_{\Lambda} (\psi_1 \wedge \dots \wedge \psi_k) \rightarrow \neg\varphi$ for some $\psi_1, \dots, \psi_k \in v_0$, from which we obtain with **(G)** and **(K)** that $\vdash_{\Lambda} \Box(\psi_1 \wedge \dots \wedge \psi_k) \rightarrow \Box\neg\varphi$. Because $\Box\psi_1 \wedge \dots \wedge \Box\psi_k \rightarrow \Box(\psi_1 \wedge \dots \wedge \psi_k)$, this implies $\vdash_{\Lambda} \Box\psi_1 \wedge \dots \wedge \Box\psi_k \rightarrow \Box\neg\varphi$. Since $\Box\psi_1, \dots, \Box\psi_k \in w$, we conclude from Lemma 1.215 that $\Box\psi_1 \wedge \dots \wedge \Box\psi_k \in w$, thus we have $\Box\neg\varphi \in w$ by modus ponens, hence $\neg\Diamond\varphi \in w$. Since w is maximal, this implies $\Diamond\varphi \notin w$. This is a contradiction. So v_0 is consistent, thus there exists by the Lindenbaum Lemma

a maximal consistent set v with $v_0 \subseteq v$. We have in particular $\varphi \in v$, and we know that $\Box\psi \in w$ implies $\psi \in v$, hence $\langle w, v \rangle \in R^\sharp$. \dashv

This helps in characterizing the model, in particular the validity relation \models by the well-known Truth Lemma.

Lemma 1.219 $\mathfrak{M}^\sharp, w \models \varphi$ iff $\varphi \in w$

Proof The proof proceeds by induction on formula φ . The statement is trivially true if $\varphi = p \in \Phi$ is a propositional letter. The set of formulas for which the assertion holds is certainly closed under Boolean operations, so the only interesting case is the case that the formula in question has the shape $\Diamond\varphi$, and that the assertion is true for φ .

“ \Rightarrow ”: If $\mathfrak{M}^\sharp, w \models \Diamond\varphi$, then we can find some v with $w R^\sharp v$ and $\mathfrak{M}^\sharp, v \models \varphi$. Thus there exists v with $\langle w, v \rangle \in R^\sharp$ such that $\varphi \in v$ by hypothesis, which in turn means $\Diamond\varphi \in w$.

“ \Leftarrow ”: Assume $\Diamond\varphi \in w$, hence there exists $v \in W^\sharp$ with $w R^\sharp v$ and $\varphi \in v$, thus $\mathfrak{M}^\sharp, v \models \varphi$. But this means $\mathfrak{M}^\sharp, w \models \Diamond\varphi$. \dashv

Finally, we obtain

Theorem 1.220 *Any normal logic is complete with respect to its canonical model.*

Proof Let Σ be a Λ -consistent set for the normal logic Λ . Then there exists by Lindenbaum’s Lemma 1.216 a maximal Λ -consistent set Σ^+ with $\Sigma \subseteq \Sigma^+$. By the Truth Lemma we have now $\mathfrak{M}^\sharp, \Sigma^+ \models \Sigma$. \dashv

1.7.3 Coalgebraic Logics

We have seen several points where coalgebras and modal logics touch each other, for example, morphisms for Kripke models are based on morphisms for the underlying \mathcal{P} -coalgebra, as a comparison of Example 1.10 and Lemma 1.193 demonstrates. Let $\mathfrak{M} = (W, R, V)$ be a Kripke model, then the accessibility relation $R \subseteq W \times W$ can be seen as a map, again denoted by R , with the signature $W \rightarrow \mathcal{P}(W)$. Map $V : \Phi \rightarrow \mathcal{P}(W)$, which indicates the validity of atomic propositions, can be decoded through a map $V_1 : W \rightarrow \mathcal{P}(\Phi)$ upon setting $V_1(w) := \{p \in \Phi \mid w \in V(p)\}$. Both V and V_1 describe the same relation $\{\langle p, w \rangle \in \Phi \times W \mid \mathfrak{M}, w \models p\}$, albeit from different angles. One can be obtained from the other one. This new representation has the advantage of describing the model from vantage point w .

Define $\mathbf{F}X := \mathcal{P}(X) \times \mathcal{P}(\Phi)$ for the set X , and put, given map $f : X \rightarrow Y$, $(\mathbf{F}f)(A, Q) := \langle f[A], Q \rangle = \langle (Pf)A, Q \rangle$ for $A \subseteq X, Q \subseteq \Phi$, then \mathbf{F} is an endofunctor on **Set**. Hence we obtain from the Kripke model \mathfrak{M} the \mathbf{F} -coalgebra (W, γ) with $\gamma(w) := R(w) \times V_1(w)$. This construction can easily be reversed: given a \mathbf{F} -coalgebra (W, γ) , we put $R(w) := \pi_1(\gamma(w))$ and $V_1(w) := \pi_2(\gamma(w))$ and construct V from V_1 , then (W, R, V) is a Kripke model (here π_1, π_2 are the projections). Thus Kripke models and \mathbf{F} -coalgebras are in an one-to-one correspondence with each other. This correspondence goes a bit deeper, as can be seen when considering morphisms.

Proposition 1.221 *Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{N} = (X, S, Y)$ be Kripke models with associated \mathfrak{F} -coalgebras (W, γ) resp. (X, δ) . Then these statements are equivalent for a map $f : W \rightarrow X$*

1. $f : (W, \gamma) \rightarrow (X, \delta)$ is a morphism of coalgebras.
2. $f : \mathfrak{M} \rightarrow \mathfrak{N}$ is a morphism of Kripke models.

Proof 1 \Rightarrow 2: We obtain for each $w \in W$ from the defining equation $(\mathbf{F}f) \circ \gamma = \delta \circ f$ these equalities

$$\begin{aligned} f[R(w)] &= S(f(w)), \\ V_1(w) &= Y_1(f(w)). \end{aligned}$$

Since $f[R(w)] = (\mathcal{P}f)(R(w))$, we conclude that $(\mathcal{P}f) \circ R = S \circ f$, so f is a morphism of the \mathcal{P} -coalgebras. We have moreover for each atomic sentence $p \in \Phi$

$$w \in V(p) \Leftrightarrow p \in V_1(w) \Leftrightarrow p \in Y_1(f(w)) \Leftrightarrow f(w) \in Y(p).$$

This means $V = f^{-1} \circ Y$, so that $f : \mathfrak{M} \rightarrow \mathfrak{N}$ is a morphism.

2 \Rightarrow 1: Because we know that $S \circ f = (\mathcal{P}f) \circ R$, and because one shows as above that $V_1 = Y_1 \circ f$, we obtain for $w \in W$

$$\begin{aligned} (\delta \circ f)(w) &= \langle S(f(w)), Y_1(f(w)) \rangle \\ &= \langle (\mathcal{P}f)(R(w)), V_1(w) \rangle \\ &= ((\mathbf{F}f) \circ \gamma)(w). \end{aligned}$$

Hence $f : (W, \gamma) \rightarrow (X, \delta)$ is a morphism for the \mathbf{F} -coalgebras. \dashv

Given a world w , the value of $\gamma(w)$ represents the worlds which are accessible from w , making sure that the validity of the atomic propositions is maintained; recall that they are not affected by a transition. This information is to be extracted in a variety of ways. We need predicate liftings for this.

Before we define liftings, however, we observe that the same mechanism works for neighborhood models.

Example 1.222 Let $\mathcal{N} = (W, N, V)$ be a neighborhood model. Define functor \mathbf{G} by putting $\mathbf{G}(X) := \mathbf{V}(X) \times \mathcal{P}(\Phi)$ for sets, and if $f : X \rightarrow Y$ is a map, put $(\mathbf{G}f)(U, Q) := \langle (\mathbf{V}f)U, Q \rangle$. Then \mathbf{G} is an endofunctor on **Set**. The \mathbf{G} -coalgebra (W, ν) associated with \mathcal{N} is defined through $\nu(w) := \langle N(w), V_1(w) \rangle$ (with V_1 defined through V as above).

Let $\mathcal{M} = (X, M, Y)$ be another neighborhood model with associated coalgebra (X, μ) . Exactly the same proof as the one for Proposition 1.221 shows that $f : \mathcal{N} \rightarrow \mathcal{M}$ is a neighborhood morphism iff $f : (W, \nu) \rightarrow (X, \mu)$ is a coalgebra morphism. \mathfrak{M}

Proceeding to define predicate liftings, let $\mathcal{P}^{\text{op}} : \mathbf{Set} \rightarrow \mathbf{Set}$ be the contravariant power set functor, i.e., given the set X , $\mathcal{P}^{\text{op}}(X)$ is the power set $\mathcal{P}(X)$ of X , and if $f : X \rightarrow Y$ is a map, then $(\mathcal{P}^{\text{op}}f) : \mathcal{P}^{\text{op}}(Y) \rightarrow \mathcal{P}^{\text{op}}(X)$ works as $B \mapsto f^{-1}[B]$.

Definition 1.223 Given a (covariant) endofunctor \mathbf{T} on **Set**, a predicate lifting λ for \mathbf{T} is a monotone natural transformation $\lambda : \mathcal{P}^{\text{op}} \rightarrow \mathcal{P}^{\text{op}} \circ \mathbf{T}$.

Interpret $A \in \mathcal{P}^{\text{op}}(X)$ as a predicate on X , then $\lambda_X(A) \in \mathcal{P}^{\text{op}}(\mathbf{T}X)$ is a predicate on $\mathbf{T}X$, hence λ_X lifts the predicate into the realm of functor \mathbf{T} ; the requirement of naturalness is intended to reflect compatibility with morphisms, as we will see below. Thus a predicate lifting helps

in specifying a requirement on the level of sets, which it then transports onto the level of those sets that are controlled by functor \mathbf{T} . Technically, this requirement means that this diagram commutes, whenever $f : X \rightarrow Y$ is a map:

$$\begin{array}{ccc} \mathcal{P}X & \xrightarrow{\lambda_X} & \mathcal{P}(\mathbf{T}X) \\ f^{-1} \uparrow & & \uparrow (\mathbf{T}f)^{-1} \\ \mathcal{P}Y & \xrightarrow{\lambda_Y} & \mathcal{P}(\mathbf{T}Y) \end{array}$$

Hence we have $\lambda_X(f^{-1}[G]) = (\mathbf{T}f)^{-1}[\lambda_Y(G)]$ for any $G \subseteq Y$.

Finally, monotonicity says that $\lambda_X(D) \subseteq \lambda_X(E)$, whenever $D \subseteq E \subseteq X$; this condition models the requirement that informations about states should only depend on their precursors. Informally it is reflected in the rule $\vdash (\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$

This example illuminates the idea.

Example 1.224 Let $\mathbf{F} = \mathcal{P}(-) \times \mathcal{P}\Phi$ be defined as above, put for the set X and for $D \subseteq X$

$$\lambda_X(D) := \{\langle D', Q \rangle \in \mathcal{P}(X) \times \mathcal{P}(\Phi) \mid D' \subseteq D\}.$$

This defines a predicate lifting $\lambda : \mathcal{P}^{\text{op}} \rightarrow \mathcal{P}^{\text{op}} \circ \mathbf{F}$. In fact, let $f : X \rightarrow Y$ be a map and $G \subseteq Y$, then


$$\begin{aligned} \lambda_X(f^{-1}[G]) &= \{\langle D', Q \rangle \mid D' \subseteq f^{-1}[G]\} \\ &= \{\langle D', Q \rangle \mid f[D'] \subseteq G\} \\ &= (\mathbf{F}f)^{-1}[\{\langle G', Q \rangle \in \mathcal{P}(Y) \times \mathcal{P}(\Phi) \mid G' \subseteq G\}] \\ &= (\mathbf{F}f)^{-1}[\lambda_Y(G)] \end{aligned}$$

(remember that $\mathbf{F}f$ leaves the second component of a pair alone). It is clear that λ_X is monotone for each set X .

Let $\gamma : W \rightarrow \mathbf{F}W$ be the coalgebra associated with Kripke model $\mathfrak{M} := (W, R, V)$, and look at this (φ is a formula)

$$\begin{aligned} w \in \gamma^{-1}[\lambda_W(\llbracket \varphi \rrbracket_{\mathfrak{M}})] &\Leftrightarrow \gamma(w) \in \lambda_W(\llbracket \varphi \rrbracket_{\mathfrak{M}}) \\ &\Leftrightarrow \langle R(w), V_1(w) \rangle \in \lambda_W(\llbracket \varphi \rrbracket_{\mathfrak{M}}) \\ &\Leftrightarrow R(w) \subseteq \llbracket \varphi \rrbracket_{\mathfrak{M}} \\ &\Leftrightarrow w \in \llbracket \Box\varphi \rrbracket_{\mathfrak{M}} \end{aligned}$$

This means that we can describe the semantics of the \Box -operator through a predicate lifting, which cooperates with the coalgebra's dynamics.

Note that it would be equally possible to do this for the \Diamond -operator: define the lifting through $D \mapsto \{\langle D', Q \rangle \mid D' \cap D \neq \emptyset\}$. But we'll stick to the \Box -operator, keeping up with tradition. 

Example 1.225 The same technique works for neighborhood models. In fact, let (W, ν) be the \mathbf{G} -coalgebra associated with neighborhood model $\mathcal{N} = (W, N, V)$ as in Example 1.222, and define

$$\lambda_X(D) := \{\langle V, Q \rangle \in \mathbf{V}(X) \times \mathcal{P}(\Phi) \mid D \in V\}.$$

Then $\lambda_X : \mathcal{P}(X) \rightarrow \mathcal{P}(\mathbf{V}(X) \times \mathcal{P}(\Phi))$ is monotone, because the elements of $\mathbf{V}X$ are upward closed. If $f : (W, \nu) \rightarrow (X, \mu)$ is a \mathbf{G} -coalgebra morphism, we obtain for $D \subseteq X$

$$\begin{aligned} \lambda_W(f^{-1}[D]) &= \{\langle V, Q \rangle \in \mathbf{V}(W) \times \mathcal{P}(\Phi) \mid f^{-1}[D] \subseteq V\} \\ &= \{\langle V, Q \rangle \in \mathbf{V}(W) \times \mathcal{P}(\Phi) \mid D \in (\mathbf{V}f)(V)\} \\ &= (\mathbf{G}f)^{-1} [\{\langle V', Q \rangle \in \mathbf{V}(X) \times \mathcal{P}(\Phi) \mid D \in V'\}] \\ &= (\mathbf{G}f)^{-1} [\lambda_X(D)] \end{aligned}$$

Consequently, λ is a predicate lifting for \mathbf{G} . We see also for formula φ

$$\begin{aligned} w \in \lambda_W(\llbracket \varphi \rrbracket_{\mathcal{N}}) &\Leftrightarrow \langle \llbracket \varphi \rrbracket_{\mathcal{N}}, V_1(w) \rangle \in \lambda_X(\llbracket \varphi \rrbracket_{\mathcal{N}}) \\ &\Leftrightarrow \llbracket \varphi \rrbracket_{\mathcal{N}} \in N(w) && \text{(by definition of } \nu) \\ &\Leftrightarrow w \in \llbracket \Box \varphi \rrbracket_{\mathcal{N}} \end{aligned}$$

Hence we can define the semantics of the \Box -operator also in this case through a predicate lifting. \uparrow

There is a general mechanism permitting us to define predicate liftings, which is outlined in the next lemma.

Lemma 1.226 *Let $\eta : \mathbf{T} \rightarrow \mathcal{P}$ be a natural transformation, and define*

$$\lambda_X(D) := \{c \in \mathbf{T}X \mid \eta_X(c) \subseteq D\}$$

for $D \subseteq X$. Then λ defines a predicate lifting for \mathbf{T} .

Proof It is clear from the construction that $D \mapsto \lambda_X(D)$ defines a monotone map, so we have to show that the diagram below is commutative for $f : X \rightarrow Y$.

$$\begin{array}{ccc} \mathcal{P}X & \xrightarrow{\lambda_X} & \mathcal{P}\mathbf{T}X \\ \uparrow f^{-1} & & \uparrow (\mathbf{T}f)^{-1} \\ \mathcal{P}Y & \xrightarrow{\lambda_Y} & \mathcal{P}\mathbf{T}Y \end{array}$$

We note that

$$\eta_X(c) \subseteq f^{-1}[E] \Leftrightarrow f[\eta_X(c)] \subseteq E \Leftrightarrow (\mathcal{P}f)(\eta_X(c)) \subseteq E$$

and

$$(\mathcal{P}f) \circ \eta_X = \eta_Y \circ (\mathbf{T}f),$$

because η is natural. Hence we obtain for $E \subseteq Y$:

$$\begin{aligned} \eta_X(f^{-1}[E]) &= \{c \in \mathbf{T}X \mid \eta_X(c) \subseteq f^{-1}[E]\} \\ &= \{c \in \mathbf{T}X \mid ((\mathcal{P}f) \circ \eta_X)(c) \subseteq E\} \\ &= \{c \in \mathbf{T}X \mid (\eta_Y \circ \mathbf{T}f)(c) \subseteq E\} \\ &= (\mathbf{T}f)^{-1} [\{d \in \mathbf{T}Y \mid \eta_Y(d) \subseteq E\}] \\ &= ((\mathbf{T}f)^{-1} \circ \eta_Y)(E). \end{aligned}$$

⊥

Let us return to the endofunctor $\mathbf{F} = \mathcal{P}(-) \times \mathcal{P}(\Phi)$ and fix for the moment an atomic proposition $p \in \Phi$. Define the constant function

$$\lambda_{p,X}(D) := \{\langle D', Q \rangle \in \mathbf{F}X \mid p \in Q\}.$$

Then an easy calculation shows that $\lambda_p : \mathcal{P}^{\text{op}} \rightarrow \mathcal{P}^{\text{op}} \circ \mathbf{F}$ is a natural transformation, hence a predicate lifting for \mathbf{F} . Let $\gamma : W \rightarrow \mathbf{F}W$ be a coalgebra with carrier W which corresponds to the Kripke model $\mathfrak{M} = (W, R, V)$, then

$$w \in (\gamma^{-1} \circ \lambda_{p,W})(D) \Leftrightarrow \gamma(w) \in \lambda_{p,W}(D) \Leftrightarrow p \in \pi_2(\gamma(w)) \Leftrightarrow w \in V(p),$$

which means that we can use λ_p for expressing the meaning of formula $p \in \Phi$. A very similar construction can be made for functor \mathbf{G} , leading to the same conclusion.

Let us cast this into a more general framework. Let $\ell_X : X \rightarrow \{0\}$ be the unique map from set X to the singleton set $\{0\}$. Given $A \subseteq \mathbf{T}(\{0\})$, define $\lambda_{A,X}(D) := \{c \in \mathbf{T}X \mid (\mathbf{T}\ell_X)(c) \in A\} = (\mathbf{T}\ell_X)^{-1}[A]$. This defines a predicate lifting for \mathbf{T} . In fact, let $f : X \rightarrow Y$ be a map, then $\ell_X = \ell_Y \circ f$, so $(\mathbf{T}f)^{-1} \circ (\mathbf{T}\ell_Y)^{-1} = ((\mathbf{T}\ell_Y) \circ (\mathbf{T}f))^{-1} = (\mathbf{T}(\ell_Y \circ f))^{-1} = (\mathbf{T}\ell_X)^{-1}$, hence $\lambda_{A,X}(f^{-1}[B]) = (\mathbf{T}f)^{-1}[\lambda_{A,Y}(B)]$. As we have seen, this construction is helpful for capturing the semantics of atomic propositions.

Negation can be treated as well in this framework. Given a predicate lifting λ for \mathbf{T} , we define for the set X and $A \subseteq X$ the set

$$\lambda_X^-(A) := (\mathbf{T}X) \setminus \lambda_X(X \setminus A),$$

then this defines a predicate lifting for \mathbf{T} . This is easily checked: monotonicity of λ^- follows from λ being monotone, and since f^{-1} is compatible with the Boolean operations, naturality follows.

Summarizing, those operations which are dear to us when interpreting modal logics through a Kripke model or through a neighborhood model can be represented using predicate liftings.

We now take a family \mathbb{L} of predicate liftings and define a logic for it.

Definition 1.227 *Let \mathbf{T} be an endofunctor on the category \mathbf{Set} of sets with maps, and let \mathbb{L} be a set of predicate listings for \mathbf{T} . The formulas for the language $\mathcal{L}(\mathbb{L})$ are defined through*

$$\varphi ::= \perp \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid [\lambda]\varphi$$

with $\varphi \in \mathbb{L}$.

The semantics of a formula in $\mathcal{L}(\mathbb{L})$ in a \mathbf{T} -coalgebra (W, γ) is defined recursively through fixing the sets of worlds $\llbracket \varphi \rrbracket_\gamma$ in which formula φ holds (with $w \models_\gamma \varphi$ iff $w \in \llbracket \varphi \rrbracket_\gamma$):

$$\begin{aligned} \llbracket \perp \rrbracket_\gamma &:= \emptyset \\ \llbracket \varphi_1 \wedge \varphi_2 \rrbracket_\gamma &:= \llbracket \varphi_1 \rrbracket_\gamma \cap \llbracket \varphi_2 \rrbracket_\gamma \\ \llbracket \neg\varphi \rrbracket_\gamma &:= W \setminus \llbracket \varphi \rrbracket_\gamma \\ \llbracket [\lambda]\varphi \rrbracket_\gamma &:= (\gamma^{-1} \circ \lambda_C)(\llbracket \varphi \rrbracket_\gamma). \end{aligned}$$

The most interesting definition is of course the last one. It is defined through a modality for the predicate lifting λ , and it says that formula $[\lambda]\varphi$ holds in world w iff the transition $\gamma(w)$ achieves a state which is lifted by λ from one in which φ holds. Hence each successor to w satisfies the predicate for φ lifted by λ .

Example 1.228 Continuing Example 1.224, we see that the simple modal logic can be defined as the modal logic for $\mathbb{L} = \{\lambda\} \cup \{\lambda_p \mid p \in \Phi\}$, where λ is defined in Example 1.224, and λ_p are the constant liftings associated with Φ . \heartsuit

We obtain also in this case the invariance of validity under morphisms.

Proposition 1.229 *Let $f : (W, \gamma) \rightarrow (X, \delta)$ be a \mathbf{T} -coalgebra morphism. Then*

$$w \models_\gamma \varphi \Leftrightarrow f(w) \models_\delta \varphi$$

holds for all formulas $\varphi \in \mathcal{L}(\mathbb{L})$ and all worlds $w \in W$.

Proof The interesting case occurs for a modal formula $[\lambda]\varphi$ with $\lambda \in \mathbb{L}$; so assume that the hypothesis is true for φ , then we have

$$\begin{aligned} f^{-1} [[[\lambda]\varphi]_\delta] &= ((\delta \circ f)^{-1} \circ \lambda_D)([\varphi]_\delta) \\ &= ((\mathbf{T}(f) \circ \gamma)^{-1} \circ \lambda_D)([\varphi]_\delta) && f \text{ is a morphism} \\ &= (\gamma^{-1} \circ (\mathbf{T}f)^{-1} \circ \lambda_D)([\varphi]_\delta) \\ &= (\gamma^{-1} \circ \lambda_C \circ f^{-1})([\varphi]_\delta) && \lambda \text{ is natural} \\ &= (\gamma^{-1} \circ \lambda_C)([\varphi]_\gamma) && \text{by hypothesis} \\ &= [[[\lambda]\varphi]_\gamma] \end{aligned}$$

\dashv

Let (C, γ) be a \mathbf{T} -coalgebra, then we define the *theory* of c

$$\text{Th}_\gamma(c) := \{\varphi \in \mathcal{L}(\mathbb{L}) \mid c \models_\gamma \varphi\}$$

for $c \in C$. Two worlds which have the same theory cannot be distinguished through formulas of the logic $\mathcal{L}(\mathbb{L})$.

Definition 1.230 *Let (C, γ) and (D, δ) be \mathbf{T} -coalgebras, $c \in C$ and $d \in D$.*

- *We call c and d are logically equivalent iff $\text{Th}_\gamma(c) = \text{Th}_\delta(d)$.*
- *The states c and d are called behaviorally equivalent iff there exists a \mathbf{T} -coalgebra (E, ϵ) and morphisms $(C, \gamma) \xrightarrow{f} (E, \epsilon) \xleftarrow{g} (D, \delta)$ such that $f(c) = g(d)$.*

Thus, logical equivalence looks locally at all the formulas which are true in a state, and then compares two states with each other. Behavioral equivalence looks for an external instance, viz., a mediating coalgebra, and at morphisms; whenever we find states the image of which coincide, we know that the states are behaviorally equivalent.

This implication is fairly easy to obtain.

Proposition 1.231 *Behaviorally equivalent states are logically equivalent.*

Proof Let $c \in C$ and $d \in D$ be behaviorally equivalent for the \mathbf{T} -coalgebras (C, γ) and (D, δ) , and assume that we have a mediating \mathbf{T} -coalgebra (E, ϵ) with morphisms

$$(C, \gamma) \xrightarrow{f} (E, \epsilon) \xleftarrow{g} (D, \delta).$$

and $f(c) = g(d)$. Then we obtain

$$\varphi \in \text{Th}_\gamma(c) \Leftrightarrow c \models_\gamma \varphi \Leftrightarrow f(c) \models_\epsilon \varphi \Leftrightarrow g(d) \models_\epsilon \varphi \Leftrightarrow d \models_\delta \varphi \Leftrightarrow \varphi \in \text{Th}_\delta(d)$$

from Proposition 1.229. \dashv

We have seen that coalgebras are useful when it comes to generalize modal logics to coalgebraic logics. Morphisms arise in a fairly natural way in this context, giving rise to defining behaviorally equivalent coalgebras. It is quite clear that bisimilarity can be treated on this level as well, by introducing a mediating coalgebra and morphisms from it; bisimilar states are logically equivalent, the argument to show this is exactly as in the case above through Proposition 1.229. In each case the question arises whether the implications can be reversed — are logically equivalent states behaviorally equivalent? Bisimilar? Answering this question requires a fairly elaborate machinery and depends strongly on the underlying functor. We will not discuss this question here but rather point to the literature, e.g., to [Pat04]. For the subprobability functor some answers and some techniques can be found in [DS11].

We finally give an idea of modelling CTL* as a popular logic for model checking coalgebraically. This shows how this modelling technique is applied, and it shows also that some additional steps become necessary, since things are not always straightforward.

Example 1.232 The logic CTL* is used for model checking [CGP99]. The abbreviation *CTL* stands for *computational tree logic*. CTL* is actually one of the simpler members of this family of tree logics used for this purpose, some of which involve continuous time [BHHK03, Dob07a]. The logic has state formulas and path formulas, the former ones are used to describe a particular state in the system, the latter ones express dynamic properties. Hence CTL* operates on two levels.

These operators are used

State operators They include the operators **A** and **E**, indicating that a property holds in a state iff it holds on all paths resp. on at least one path emanating from it,

Path operators They include the operators

- **X** for *next time* — a property holds in the next, i.e., second state of a path,
- **F** for *in the future* — the specified property holds for some state on the path,
- **G** for *globally* — the property holds always on a path,
- **U** for *until* — this requires two properties as arguments; it holds on a path if there exists a state on the path for which the second property holds, and the first one holds on each preceding state.

State formulas are given through this syntax:

$$\varphi ::= \perp \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{E}\psi \mid \mathbf{A}\psi$$

with $p \in \Phi$ an atomic proposition and ψ a path formula. Path formulas are given through

$$\psi ::= \varphi \mid \neg\psi \mid \psi_1 \wedge \psi_2 \mid \mathbf{X}\psi \mid \mathbf{F}\psi \mid \mathbf{G}\psi \mid \psi_1 \mathbf{U}\psi_2$$

with φ a state formula. So both state and path formulas are closed under the usual Boolean operations, each atomic proposition is a state formula, and state formulas are also path formulas. Path formulas are closed under the operators $\mathbf{X}, \mathbf{F}, \mathbf{G}, \mathbf{U}$, and the operators \mathbf{A} and \mathbf{E} convert a path formula to a state formula.

Let W be the set of all states, and assume that $V : \Phi \rightarrow \mathcal{P}(W)$ assigns to each atomic formula the states for which it is valid. We assume also that we are given a transition relation $R \subseteq W \times W$; it is sometimes assumed that R is left total, but this is mostly for computational reasons, so we will not make this assumption here. Put

$$S := \{ \langle w_1, w_2, \dots \rangle \in W^\infty \mid w_i R w_{i+1} \text{ for all } i \in \mathbb{N} \}$$

as the set of all infinite R -paths over W . The interpretation of formulas is then defined as follows:

State formulas Let $w \in W$, $\varphi, \varphi_1, \varphi_2$ be state formulas and ψ be a path formula, then

$$\begin{aligned} w \models \top &\Leftrightarrow \text{always} \\ s \models p &\Leftrightarrow w \in V(p) \\ w \models \neg\varphi &\Leftrightarrow w \models \varphi \text{ is false} \\ w \models \varphi_1 \wedge \varphi_2 &\Leftrightarrow w \models \varphi_1 \text{ and } w \models \varphi_2 \\ w \models \mathbf{E}\psi &\Leftrightarrow \sigma \models \psi \text{ for some path } \sigma \text{ starting from } w \\ w \models \mathbf{A}\psi &\Leftrightarrow \sigma \models \psi \text{ for all paths } \sigma \text{ starting from } w \end{aligned}$$

Path formulas Let $\sigma \in S$ be an infinite path with first node σ_1 , σ^k is the path with the first k nodes deleted; ψ is a path formula, and φ a state formula, then

$$\begin{aligned} \sigma \models \varphi &\Leftrightarrow \sigma_1 \models \varphi \\ \sigma \models \neg\psi &\Leftrightarrow \sigma \models \psi \text{ is false} \\ \sigma \models \psi_1 \wedge \psi_2 &\Leftrightarrow \sigma \models \psi_1 \text{ and } \sigma \models \psi_2 \\ \sigma \models \mathbf{X}\psi &\Leftrightarrow \sigma^1 \models \psi \\ \sigma \models \mathbf{F}\psi &\Leftrightarrow \sigma^k \models \psi \text{ for some } k \geq 0 \\ \sigma \models \mathbf{G}\psi &\Leftrightarrow \sigma^k \models \psi \text{ for all } k \geq 0 \\ \sigma \models \psi_1 \mathbf{U}\psi_2 &\Leftrightarrow \exists k \geq 0 : \sigma^k \models \psi_2 \text{ and } \forall 0 \leq j < k : \sigma^j \models \psi_1. \end{aligned}$$

Thus a state formula holds on a path iff it holds on the first node, $\mathbf{X}\psi$ holds on path σ iff ψ holds on σ with its first node deleted, and $\psi_1 \mathbf{U}\psi_2$ holds on path σ iff ψ_2 holds on σ^k for some k , and iff ψ_1 holds on σ^i for all i preceding k .

We would have to provide interpretations only for conjunction, negation, for \mathbf{A} , \mathbf{X} , and \mathbf{U} . This is so since \mathbf{E} is the nabla of \mathbf{A} , \mathbf{G} is the nabla of \mathbf{F} , and $\mathbf{F}\psi$ is equivalent to $(\neg\perp)\mathbf{U}\psi$.

Conjunction and negation are easily interpreted, so we have to take care only of the temporal operators **A**, **X** and **U**.

A coalgebraic interpretation reads as follows. The \mathcal{P} -coalgebras together with their morphisms form a category **CoAlg**. Let (X, R) be a \mathcal{P} -coalgebra, then

$$\mathbf{R}(X, R) := \{(x_n)_{n \in \mathbb{N}} \in X^\infty \mid x_n R x_{n+1} \text{ for all } n \in \mathbb{N}\}$$

is the object part of a functor, $(\mathbf{R}f)((x_n)_{n \in \mathbb{N}}) := (f(x_n)_{n \in \mathbb{N}})$ sends each coalgebra morphism $f : (X, R) \rightarrow (Y, S)$ to a map $(\mathbf{R}f) : \mathbf{R}(X, R) \rightarrow \mathbf{R}(Y, S)$, which maps $(x_n)_{n \in \mathbb{N}}$ to $f(x_n)_{n \in \mathbb{N}}$; recall that $x R x'$ implies $f(x) S f(x')$. Thus $\mathbf{R} : \mathbf{CoAlg} \rightarrow \mathbf{Set}$ is a functor. Note that the transition structure of the underlying Kripke model is already encoded through functor \mathbf{R} . This is reflected in the definition of the dynamics $\gamma : X \rightarrow \mathbf{R}(X, R) \times \mathcal{P}(\Phi)$ upon setting

$$\gamma(x) := \langle \{w \in \mathbf{R}(X, R) \mid w_1 = x\}, V_1(x) \rangle,$$

where $V_1 : X \rightarrow \mathcal{P}(\Phi)$ is defined according to $V : \Phi \rightarrow \mathcal{P}(X)$ as above. Define for the model $\mathfrak{M} := (W, R, V)$ the map $\lambda_{\mathbf{R}(W, R)} : C \mapsto \{\langle C', A \rangle \in \mathcal{P}(\mathbf{R}(W, R)) \times \mathcal{P}(\Phi) \mid C' \subseteq C\}$, then λ defines a natural transformation $\mathcal{P}^{\text{op}} \circ \mathbf{R} \rightarrow \mathcal{P}^{\text{op}} \circ \mathbf{F} \circ \mathbf{R}$ (the functor \mathbf{F} has been defined in Example 1.224); note that we have to check naturality in terms of model morphisms, which are in particular morphisms for the underlying \mathcal{P} -coalgebra. Thus we can define for $w \in W$

$$w \models_{\mathfrak{M}} \mathbf{A}\psi \Leftrightarrow w \in \gamma^{-1} \circ \lambda_{\mathbf{R}(W, R)}(\llbracket \psi \rrbracket_{\mathfrak{M}})$$

In a similar way we define $w \models_{\mathfrak{M}} p$ for atomic propositions $p \in \Phi$; this is left to the reader.

The interpretation of path formulas requires a slightly different approach. We define

$$\begin{aligned} \mu_{\mathbf{R}(X, R)}(A) &:= \{\sigma \in \mathbf{R}(X, R) \mid \sigma^1 \in A\}, \\ \vartheta_{\mathbf{R}(X, R)}(A, B) &:= \bigcup_{k \in \mathbb{N}} \{\sigma \in \mathbf{R}(X, R) \mid \sigma^k \in B, \sigma^i \in A \text{ for } 0 \leq i < k\}, \end{aligned}$$

whenever $A, B \in \mathbf{R}(X, R)$. Then $\mu : \mathcal{P}^{\text{op}} \circ \mathbf{R} \rightarrow \mathcal{P}^{\text{op}} \circ \mathbf{R}$ and $\vartheta : (\mathcal{P}^{\text{op}} \circ \mathbf{R}) \times (\mathcal{P}^{\text{op}} \circ \mathbf{R}) \rightarrow \mathcal{P}^{\text{op}} \circ \mathbf{R}$ are natural transformations, and we put

$$\begin{aligned} \llbracket \mathbf{X}\psi \rrbracket_{\mathfrak{M}} &:= \mu_{\mathbf{R}(M, R)}(\llbracket \psi \rrbracket_{\mathfrak{M}}), \\ \llbracket \psi_1 \mathbf{U} \psi_2 \rrbracket_{\mathfrak{M}} &:= \vartheta_{\mathbf{R}(X, R)}(\llbracket \psi_1 \rrbracket_{\mathfrak{M}}, \llbracket \psi_2 \rrbracket_{\mathfrak{M}}). \end{aligned}$$

The example shows that a two level logics can be interpreted as well through a coalgebraic approach, provided the predicate liftings which characterize this approach are complemented by additional natural transformations (which are called *bridge operators* in [Dob09]) \mathfrak{M}

1.8 Bibliographic Notes

The monograph by Mac Lane [Lan97] discusses all the definitions and basic constructions; the text [BW99] takes much of its motivation for categorical constructions from applications in computer science. Monads are introduced following essentially Moggi's seminal paper [Mog91]. The text book [Pum99] is an exposition fine tuned towards students interested in categories;

the proof of Lemma 1.82 and the discussion on Yoneda's construction follows its exposition rather closely. The discrete probability functor has been studied extensively in [Sok05], its continuous step twin in [Gir81, Dob07b]. The use of upper closed subsets for the interpretation of game logic is due to Parikh [Par85], [PP03] defines bisimilarity in this context. The coalgebraic interpretation is investigated in [Dob10]. Coalgebras are carefully discussed at length in [Rut00], by which the present discussion has been inspired.

The programming language `Haskell` is discussed in a growing number of accessible books, a personal selection includes [OGS09, Lip11]; the present short discussion is taken from [Dob12]. The representation of modal logics draws substantially from [BdRV01], and the discussion on coalgebraic logic is strongly influenced by [Pat04] and by the survey paper [DS11] as well as the monograph [Dob09].

1.9 Exercises

Exercise 1 The category **uGraph** has as objects undirected graphs. A morphism $f : (G, E) \rightarrow (H, F)$ is a map $f : G \rightarrow H$ such that $\{f(x), f(y)\} \in F$ whenever $\{x, y\} \in E$ (hence a morphism respects edges). Show that the laws of a category are satisfied.

Exercise 2 A morphism $f : a \rightarrow b$ in a category **K** is a *split monomorphism* iff it has a left inverse, i.e. there exists $g : b \rightarrow a$ such that $g \circ f = \text{id}_a$. Similarly, f is a *split epimorphism* iff it has a right inverse, i.e. there exists $g : b \rightarrow a$ such that $f \circ g = \text{id}_b$.

1. Show that every split monomorphism is monic and every split epimorphism is epic.
2. Show that a split epimorphism that is monic must be an isomorphism.
3. Show that for a morphism $f : a \rightarrow b$, it holds that:
 - (i) f is a split monomorphism $\Leftrightarrow \text{hom}_{\mathbf{K}}(f, x)$ is surjective for every object x ,
 - (ii) f is a split epimorphism $\Leftrightarrow \text{hom}_{\mathbf{K}}(x, f)$ is surjective for every object x ,
4. Characterize the split monomorphisms in **Set**. What can you say about split epimorphisms in **Set**?

Exercise 3 The category **Par** of sets and partial functions is defined as follows:

- Objects are sets.
- A morphism in $\text{hom}_{\mathbf{Par}}(A, B)$ is a partial function $f : A \rightharpoonup B$, i.e. it is a set-theoretic function $f : \text{car}(f) \rightarrow B$ from a subset $\text{car}(f) \subseteq A$ into B . $\text{car}(f)$ is called the *carrier* of f .
- The identity $\text{id}_A : A \rightharpoonup A$ is the usual identity function with $\text{car}(\text{id}_A) = A$.
- For $f : A \rightharpoonup B$ and $g : B \rightharpoonup C$ the composition $g \circ f$ is defined as the usual composition $g(f(x))$ on the carrier:

$$\text{car}(g \circ f) := \{x \in \text{car}(f) \mid f(x) \in \text{car}(g)\}.$$

1. Show that **Par** is a category and characterize its monomorphisms and epimorphisms.
2. Show that the usual set-theoretic Cartesian product you know is not the categorical product in **Par**. Characterise binary products in **Par**.

Exercise 4 Define the category **Pos** of ordered sets and monotone maps. The objects are ordered sets (P, \leq) , morphisms are monotone maps $f : (P, \leq) \rightarrow (Q, \sqsubseteq)$, i.e. maps $f : P \rightarrow Q$ such that $x \leq y$ implies $f(x) \sqsubseteq f(y)$. Composition and identities are inherited from **Set**.

1. Show that under this definition **Pos** is a category.
2. Characterize monomorphisms and epimorphisms in **Pos**.
3. Give an example of an ordered set (P, \leq) which is isomorphic (in **Pos**) to $(P, \leq)^{\text{op}}$ but $(P, \leq) \neq (P, \leq)^{\text{op}}$.

An ordered set (P, \leq) is called *totally ordered* if for all $x, y \in P$ it holds that $x \leq y$ or $y \leq x$.

Show that if (P, \leq) is isomorphic (in **Pos**) to a totally ordered set (Q, \sqsubseteq) , then (P, \leq) is also totally ordered. Use this result to give an example of a monotone map $f : (P, \leq) \rightarrow (Q, \sqsubseteq)$ that is monic and epic but not an isomorphism.

Exercise 5 Given a set X , the set of (finite) strings of elements of X is again denoted by X^* .

1. Show that X^* forms a monoid under concatenation, the *free monoid* over X .
2. Given a map $f : X \rightarrow Y$, extend it uniquely to a monoid morphism $f^* : X^* \rightarrow Y^*$. In particular for all $x \in X$, it should hold that $f^*(\langle x \rangle) = \langle f(x) \rangle$, where $\langle x \rangle$ denotes the string consisting only of the character x .
3. Under what conditions on X is X^* a commutative monoid, i.e. has a commutative operation?

Exercise 6 Let $(M, *)$ be a monoid. We define a category **M** as follows: it has only one object $*$, $\text{hom}_{\mathbf{M}}(*, *) = M$ with id_* as the unit of the monoid, and composition is defined through $m_2 \circ m_1 := m_2 * m_1$.

1. Show that **M** indeed forms a category.
2. Characterize the dual category \mathbf{M}^{op} . When are **M** and \mathbf{M}^{op} equal?
3. Characterize monomorphisms, epimorphisms and isomorphisms for finite M . (What happens in the infinite case?)

Exercise 7 Let (S, \mathcal{A}) and (T, \mathcal{B}) be measurable spaces, and assume that the σ -algebra \mathcal{B} is generated by \mathcal{B}_0 . Show that a map $f : S \rightarrow T$ is \mathcal{A} - \mathcal{B} -measurable iff $f^{-1}[B_0] \in \mathcal{A}$ for all $B_0 \in \mathcal{B}_0$.

Exercise 8 Let (S, \mathcal{A}) and (T, \mathcal{B}) be measurable spaces and $f : S \rightarrow T$ be \mathcal{A} - \mathcal{B} -measurable. Define $f_*(\mu)(B) := \mu(f^{-1}[B])$ for $\mu \in \mathbb{S}(S, \mathcal{A})$, $B \in \mathcal{B}$, then $f_* : \mathbb{S}(S, \mathcal{A}) \rightarrow \mathbb{S}(T, \mathcal{B})$. Show that f_* is $w(\mathcal{A})$ - $w(\mathcal{B})$ -measurable. Hint: Use Exercise 7.

Exercise 9 Let S be a countable sets with $p : S \rightarrow [0, 1]$ as a *discrete probability distribution*, thus $\sum_{s \in S} p(s) = 1$; denote the corresponding probability measure on $\mathcal{P}(S)$ by μ_p , hence $\mu_p(A) = \sum_{s \in A} p(s)$. Let T be an at most countable set with a discrete probability distribution q . Show that a map $f : S \rightarrow T$ is a morphism for the probability spaces $(S, \mathcal{P}(S), \mu_p)$ and $(T, \mathcal{P}(T), \mu_q)$ iff $q(t) = \sum_{f(s)=t} p(s)$ holds for all $t \in T$.

Exercise 10 Show that $\{x \in [0, 1] \mid \langle x, x \rangle \in E\} \in \mathcal{B}([0, 1])$, whenever $E \in \mathcal{B}([0, 1]) \otimes \mathcal{B}([0, 1])$.

Exercise 11 Let's chase some diagrams. Consider the following diagram in a category \mathbf{K} :

$$\begin{array}{ccccc}
 a & \xrightarrow{f} & b & \xrightarrow{g} & c \\
 k \downarrow & & \downarrow \ell & & \downarrow m \\
 x & \xrightarrow{r} & y & \xrightarrow{s} & z
 \end{array}$$

1. Show that if the left inner and right inner diagrams commute, then the outer diagram commutes as well.
2. Show that if the outer and right inner diagrams commute and s is a monomorphism, then the left inner diagram commutes as well.
3. Give examples in **Set** such that:
 - (a) the outer and left inner diagrams commute, but not the right inner diagram,
 - (b) the outer and right inner diagrams commute, but not the left inner diagram.

Exercise 12 Give an example of a product in a category \mathbf{K} such that one of the projections is not epic.

Exercise 13 What can you say about products and sums in the category \mathbf{M} given by a finite monoid $(M, *)$, as defined in Exercise 5? (Consider the case that $(M, *)$ is commutative first.)

Exercise 14 Show that the product topology has this universal property: $f : (D, \mathcal{D}) \rightarrow (S \times T, \mathcal{G} \times \mathcal{H})$ is continuous iff $\pi_S \circ f : (D, \mathcal{D}) \rightarrow (S, \mathcal{G})$ and $\pi_T \circ f : (D, \mathcal{D}) \rightarrow (T, \mathcal{H})$ are continuous. Formulate and prove the corresponding property for morphisms in **Meas**.

Exercise 15 A collection of morphisms $\{f_i : a \rightarrow b_i\}_{i \in I}$ with the same domain in category \mathbf{K} is called *jointly monic* whenever the following holds: If $g_1 : x \rightarrow a$ and $g_2 : x \rightarrow a$ are morphisms such that $f_i \circ g_1 = f_i \circ g_2$ for all $i \in I$, then $g_1 = g_2$. Dually one defines a collection of morphisms to be *jointly epic*.

Show that the projections from a categorical product are jointly monic and the injections into a categorical sum are jointly epic.

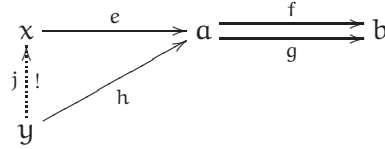
Exercise 16 Assume the following diagram in a category \mathbf{K} commutes:

$$\begin{array}{ccccc}
 a & \xrightarrow{f} & b & \xrightarrow{g} & c \\
 k \downarrow & & \downarrow \ell & & \downarrow m \\
 x & \xrightarrow{r} & y & \xrightarrow{s} & z
 \end{array}$$

Prove or disprove: if the outer diagram is a pullback, one of the inner diagrams is a pullback as well. Which inner diagram has to be a pullback for the outer one to be also a pullback?

Exercise 17 Suppose $f, g : a \rightarrow b$ are morphisms in a category \mathbf{C} . An *equalizer* of f and g is a morphism $e : x \rightarrow a$ such that $f \circ e = g \circ e$, and whenever $h : y \rightarrow a$ is a morphism with $f \circ h = g \circ h$, then there exists a unique $j : y \rightarrow x$ such that $h = e \circ j$.

This is the diagram:



1. Show that equalizers are uniquely determined up to isomorphism.
2. Show that the morphism $e : x \rightarrow a$ is a monomorphism.
3. Show that a category has pullbacks if it has products and equalizers.

Exercise 18 A *terminal object* in category \mathbf{K} is an object $\mathbf{1}$ such that for every object a there exists a unique morphism $! : a \rightarrow \mathbf{1}$.

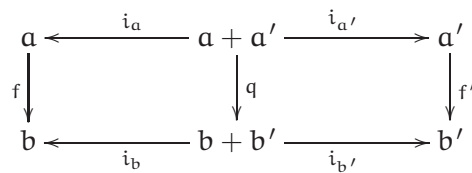
1. Show that terminal objects are uniquely determined up to isomorphism.
2. Show that a category has (binary) products and equalizers if it has pullbacks and a terminal object.

Exercise 19 Show that the coproduct σ -algebra has this universal property: $f : (S + T, \mathcal{A} + \mathcal{B}) \rightarrow (R, \mathcal{X})$ is $\mathcal{A} + \mathcal{B}$ - \mathcal{X} -measurable iff $f \circ i_S$ and $f \circ i_T$ are \mathcal{A} - \mathcal{X} - resp. \mathcal{B} - \mathcal{X} -measurable. Formulate and prove the corresponding property for morphisms in **Top**.

Exercise 20 Assume that in category \mathbf{K} any two elements have a product. Show that $a \times (b \times c)$ and $(a \times b) \times c$ are isomorphic.

Exercise 21 Prove Lemma 1.54.

Exercise 22 Assume that the coproducts $a + a'$ and $b + b'$ exist in category \mathbf{K} . Given morphisms $f : a \rightarrow b$ and $f' : a' \rightarrow b'$, show that there exists a unique morphism $q : a + a' \rightarrow b + b'$ such that this diagram commutes



Exercise 23 Show that the category **Prob** has no coproducts (Hint: Considering $(S, \mathcal{C}) + (T, \mathcal{D})$, show that, e.g., $i_S^{-1}[i_S[A]]$ equals A for $A \subseteq S$).

Exercise 24 Identify the product of two objects in the category **Rel** of relations.

Exercise 25 We investigate the epi-mono factorization in the category **Meas** of measurable spaces. Fix two measurable spaces (S, \mathcal{A}) and (T, \mathcal{B}) and a morphism $f : (S, \mathcal{A}) \rightarrow (T, \mathcal{B})$.

1. Let $\mathcal{A}/\ker(f)$ be the largest σ -algebra \mathcal{X} on $S/\ker(f)$ rendering the factor map $\eta_{\ker(f)} : S \rightarrow S/\ker(f)$ \mathcal{A} - \mathcal{X} -measurable. Show that $\mathcal{A}/\ker(f) = \{C \subseteq S/\ker(f) \mid \eta_{\ker(f)}^{-1}[C] \in \mathcal{A}\}$, and show that $\mathcal{A}/\ker(f)$ has this universal property: given a measurable space (Z, \mathcal{C}) , a map $g : S/\ker(f) \rightarrow Z$ is $\mathcal{A}/\ker(f)$ - \mathcal{C} measurable iff $g \circ \eta_{\ker(f)} : S \rightarrow Z$ is \mathcal{A} - \mathcal{C} -measurable.

2. Show that $\eta_{\ker(f)}$ is an epimorphism in **Meas**, and that $f_\bullet : [x]_{\ker(f)} \mapsto f(x)$ is a monomorphism in **Meas**.
3. Let $f = m \circ e$ with an epimorphism $e : (S, \mathcal{A}) \rightarrow (Z, \mathcal{C})$ and a monomorphism $m : (Z, \mathcal{C}) \rightarrow (T, \mathcal{B})$, and define $b : S/\ker(f) \rightarrow Z$ through $[s]_{\ker(f)} \mapsto e(s)$, see Corollary 1.27. Show that b is $\mathcal{A}/\ker(f)$ - \mathcal{C} -measurable, and prove or disprove measurability of b^{-1} .

Exercise 26 Let **AbGroup** be the category of Abelian groups. Its objects are commutative groups, a morphism $\varphi : (G, +) \rightarrow (H, *)$ is a map $\varphi : G \rightarrow H$ with $\varphi(a + b) = \varphi(a) * \varphi(b)$ and $\varphi(-a) = -\varphi(a)$. Each subgroup V of an Abelian group $(G, *)$ defines an equivalence relation ρ_V through $a \rho_V b$ iff $a - b \in V$. Characterize the pushout of η_{ρ_V} and η_{ρ_W} for subgroups V and W in **AbGroup**.

Exercise 27 Given a set X , define $\mathbf{F}(X) := X \times X$, for a map $f : X \rightarrow Y$, $\mathbf{F}(f)(x_1, x_2) := \langle f(x_1), f(x_2) \rangle$ is defined. Show that \mathbf{F} is an endofunctor on **Set**.

Exercise 28 Fix a set A of labels; define $\mathbf{F}(X) := \{*\} \cup A \times X$ for the set X , if $f : X \rightarrow Y$ is a map, put $\mathbf{F}(f)(*) := *$ and $\mathbf{F}(f)(a, x) := \langle a, f(x) \rangle$. Show that $\mathbf{F} : \mathbf{Set} \rightarrow \mathbf{Set}$ defines an endofunctor.

This endofunctor models termination or labeled output.

Exercise 29 Fix a set A of labels, and put for the set X

$$\mathbf{F}(X) := \mathcal{P}_f(A \times X),$$

where \mathcal{P}_f denotes all finite subsets of its argument. Thus $G \subseteq \mathbf{F}(X)$ is a finite subset of $A \times X$, which models finite branching, with $\langle a, x \rangle \in G$ as one of the possible branches, which is in this case labeled by $a \in A$. Define

$$\mathbf{F}(f)(B) := \{\langle a, f(x) \rangle \mid \langle a, x \rangle \in B\}$$

for the map $f : X \rightarrow Y$ and $B \subseteq A \times X$. Show that $\mathbf{F} : \mathbf{Set} \rightarrow \mathbf{Set}$ is an endofunctor.

Exercise 30 Show that the limit cone for a functor $\mathbf{F} : \mathbf{K} \rightarrow \mathbf{L}$ is unique up to isomorphisms, provided it exists.

Exercise 31 Let $I \neq \emptyset$ be an arbitrary index set, and let \mathbf{K} be the discrete category over I . Given a family $(X_i)_{i \in I}$, define $\mathbf{F} : I \rightarrow \mathbf{Set}$ by $\mathbf{F}i := X_i$. Show that

$$X := \prod_{i \in I} X_i := \{x : I \rightarrow \bigcup_{i \in I} X_i \mid x(i) \in X_i \text{ for all } i \in I\}$$

with $\pi_i : x \mapsto x(i)$ is a limit $(X, (\pi_i)_{i \in I})$ of \mathbf{F} .

Exercise 32 Formulate the equalizer of two morphisms (cp. Exercise 17) as a limit.

Exercise 33 Define for the set X the free monoid X^* generated by X through

$$X^* := \{\langle x_1, \dots, x_k \rangle \mid x_i \in X, k \geq 0\}$$

with juxtaposition as multiplication, i.e., $\langle x_1, \dots, x_k \rangle * \langle x'_1, \dots, x'_r \rangle := \langle x_1, \dots, x_k, x'_1, \dots, x'_r \rangle$; the neutral element ϵ is $\langle x_1, \dots, x_k \rangle$ with $k = 0$. Define

$$\begin{aligned} f^*(x_1 * \dots * x_k) &:= f(x_1) * \dots * f(x_k) \\ \eta_X(x) &:= \langle x \rangle \end{aligned}$$

for the map $f : X \rightarrow Y^*$ and $x \in X$. Put $\mathbf{F}X := X^*$. Show that $(\mathbf{F}, \eta, -^*)$ is a Kleisli triple, and compare it with the `list` monad, see page 52. Compute μ_x for this monad.

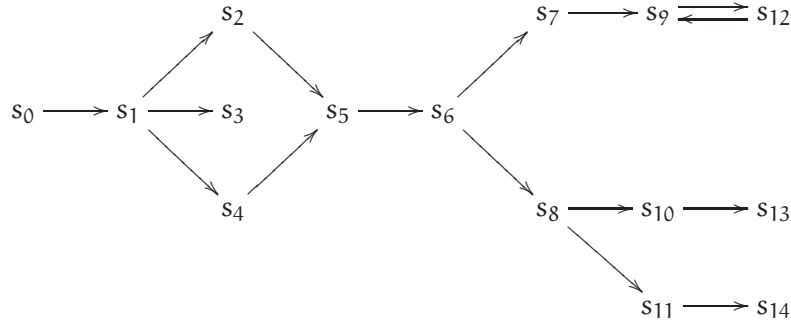
Exercise 34 Given are the systems S and T .



- Consider the transition systems S and T as coalgebras for a suitable functor $\mathbf{F} : \mathbf{Set} \rightarrow \mathbf{Set}$, $X \mapsto \mathcal{P}(X)$. Determine the dynamics of the respective coalgebras.
- Show that there is no coalgebra morphism $S \rightarrow T$.
- Construct a coalgebra morphism $T \rightarrow S$.
- Construct a bisimulation between S and T as a coalgebra on the carrier

$$\{\langle s_2, t_3 \rangle, \langle s_2, t_4 \rangle, \langle s_4, t_2 \rangle, \langle s_5, t_6 \rangle, \langle s_5, t_7 \rangle, \langle s_6, t_5 \rangle\}.$$

Exercise 35 Characterize this nondeterministic transition system S as a coalgebra for a suitable functor $\mathbf{F} : \mathbf{Set} \rightarrow \mathbf{Set}$.



Show that

$$\alpha := \{\langle s_i, s_i \rangle \mid 0 \leq i \leq 12\} \cup \{\langle s_2, s_4 \rangle, \langle s_4, s_2 \rangle, \langle s_9, s_{12} \rangle, \langle s_{12}, s_9 \rangle, \langle s_{13}, s_{14} \rangle, \langle s_{14}, s_{13} \rangle\}$$

is a bisimulation equivalence on S . Simplify S by giving a coalgebraic characterisation of the factor system S/α . Furthermore, determine whether α is the largest bisimulation equivalence on S .

Exercise 36 The deterministic finite automata A_1, A_2 with input and output alphabet $\{0, 1\}$

and the following transition tables are given:

A_1	state	input	output	next state	A_2	state	input	output	next state
	s_0	0	0	s_1		s'_0	0	0	s'_0
	s_0	1	1	s_0		s'_0	1	1	s'_1
	s_1	0	0	s_2		s'_1	0	0	s'_0
	s_1	1	1	s_3		s'_1	1	1	s'_2
	s_2	0	1	s_4		s'_2	0	1	s'_3
	s_2	1	0	s_2		s'_2	1	0	s'_2
	s_3	0	0	s_1		s'_3	0	1	s'_4
	s_3	1	1	s_3		s'_3	1	0	s'_2
	s_4	0	1	s_3		s'_4	0	0	s'_5
	s_4	1	0	s_2		s'_4	1	1	s'_4
						s'_5	0	0	s'_2
						s'_5	1	1	s'_4

1. Formalize the automata as coalgebras for a suitable functor $\mathbf{F} : \mathbf{Set} \rightarrow \mathbf{Set}$, $\mathbf{F}(X) = (X \times O)^I$. (You have to choose I and O first.)
2. Construct a coalgebra morphism from A_1 to A_2 and use this to find a bisimulation R between A_1 and A_2 . Describe the dynamics of R coalgebraically.

Exercise 37 Let P be an effectivity function on X , and define $\partial P(A) := X \setminus P(X \setminus A)$. Show that ∂P defines an effectivity function on X . Given an effectivity function Q on Y and a morphism $f : P \rightarrow Q$, show that $f : \partial P \rightarrow \partial Q$ is a morphism as well.

Exercise 38 Show that the power set functor $\mathcal{P} : \mathbf{Set} \rightarrow \mathbf{Set}$ does not preserve pullbacks. (Hint: You can use the fact, that in \mathbf{Set} the pullback of the left diagram is explicitly given as $P := \{\langle x, y \rangle \mid f(x) = g(y)\}$ with π_X and π_Y being the usual projections.)

Exercise 39 Suppose $\mathbf{F}, \mathbf{G} : \mathbf{Set} \rightarrow \mathbf{Set}$ are functors.

1. Show that if \mathbf{F} and \mathbf{G} both preserve weak pullbacks, then also the product functor $\mathbf{F} \times \mathbf{G} : \mathbf{Set} \rightarrow \mathbf{Set}$, defined as $(\mathbf{F} \times \mathbf{G})(X) = \mathbf{F}(X) \times \mathbf{G}(X)$ and $(\mathbf{F} \times \mathbf{G})(f) = \mathbf{F}(f) \times \mathbf{G}(f)$, preserves weak pullbacks.
2. Generalize to arbitrary products, i.e show the following: If I is a set and for every $i \in I$, $\mathbf{F}_i : \mathbf{Set} \rightarrow \mathbf{Set}$ is a functor preserving weak pullbacks, then also the product functor $\prod_{i \in I} \mathbf{F}_i : \mathbf{Set} \rightarrow \mathbf{Set}$ preserves pullbacks.

Use this to show that the exponential functor $(-)^A : \mathbf{Set} \rightarrow \mathbf{Set}$, given by $X \mapsto X^A = \prod_{a \in A} X$ and $f \mapsto f^A = \prod_{a \in A} f$ preserves weak pullbacks.

3. Show that if \mathbf{F} preserves weak pullbacks and there exist natural transformations $\eta : \mathbf{F} \rightarrow \mathbf{G}$ and $\nu : \mathbf{G} \rightarrow \mathbf{F}$, then also \mathbf{G} preserves weak pullbacks.
4. Show that if both \mathbf{F} and \mathbf{G} preserve weak pullbacks, then also $\mathbf{F} + \mathbf{G} : \mathbf{Set} \rightarrow \mathbf{Set}$, defined as $X \mapsto \mathbf{F}(X) + \mathbf{G}(X)$ and $f \mapsto \mathbf{F}(f) + \mathbf{G}(f)$, preserves weak pullbacks. (Hint: Show first that for every morphism $f : X \rightarrow A + B$, one has a decomposition $X \cong X_A + X_B$ and $f_A : X_A \rightarrow A$, $f_B : X_B \rightarrow B$ such that $f \cong (f_A \circ i_A) + (f_B \circ i_B)$.)

Exercise 40 Consider the modal similarity type $\tau = (O, \rho)$, with $O := \{\langle a \rangle, \langle b \rangle\}$ and $\rho(\langle a \rangle) = \rho(\langle b \rangle) = 1$, over the propositional letters $\{p, q\}$. Let furthermore $[a], [b]$ denote the nablas of $\langle a \rangle$ and $\langle b \rangle$.

Show that the following formula is a tautology, i.e. it holds in every possible τ -model:

$$(\langle a \rangle p \vee \langle a \rangle q \vee [b](\neg p \vee q)) \rightarrow (\langle a \rangle(p \vee q) \vee \neg[b]p \vee [b]q)$$

A *frame morphism* between frames $(X, (R_{\langle a \rangle}, R_{\langle b \rangle}))$ and $(Y, (S_{\langle a \rangle}, S_{\langle b \rangle}))$ is given for this modal similarity type by a map $f : X \rightarrow Y$ which satisfies the following properties:

- If $\langle x, x_1 \rangle \in R_{\langle a \rangle}$, then $\langle f(x), f(x_1) \rangle \in S_{\langle a \rangle}$. Moreover, if $\langle f(x), y_1 \rangle \in S_{\langle a \rangle}$, then there exists $x_1 \in X$ with $\langle x, x_1 \rangle \in R_{\langle a \rangle}$ and $y_1 = f(x_1)$.
- If $\langle x, x_1 \rangle \in R_{\langle b \rangle}$, then $\langle f(x), f(x_1) \rangle \in S_{\langle b \rangle}$. Moreover, if $\langle f(x), y_1 \rangle \in S_{\langle b \rangle}$, then there exists $x_1 \in X$ with $\langle x, x_1 \rangle \in R_{\langle b \rangle}$ and $y_1 = f(x_1)$.

Give a coalgebraic definition of frame morphisms for this modal similarity type, i.e. find a functor $\mathbf{F} : \mathbf{Set} \rightarrow \mathbf{Set}$ such that frame morphisms correspond to \mathbf{F} -coalgebra morphisms.

Exercise 41 Consider the fragment of PDL defined mutually recursive by:

Formulas $\varphi ::= p \mid \varphi_1 \wedge \varphi_2 \mid \neg \varphi_1 \mid \langle \pi \rangle \varphi$ (where $p \in \Phi$ for a set of basic propositions Φ , and π is a program).

Programs $\pi ::= t \mid \pi_1; \pi_2 \mid \varphi?$ (where $t \in \text{Bas}$ for a set of basic programs Bas and φ is a formula).

Suppose you are given the set of basic programs $\text{Bas} := \{\text{init}, \text{run}, \text{print}\}$ and basic propositions $\Phi := \{\text{is_init}, \text{did_print}\}$.

We define a model \mathfrak{M} for this language as follows:

- The basic set of \mathfrak{M} is $X := \{-1, 0, 1\}$.
- The modal formulas for basic programs are interpreted by the relations

$$\begin{aligned} R_{\text{init}} &:= \{\langle -1, 0 \rangle, \langle 0, 0 \rangle, \langle 1, 1 \rangle\}, \\ R_{\text{run}} &:= \{\langle -1, -1 \rangle, \langle 0, 0 \rangle, \langle 1, 1 \rangle\}, \\ R_{\text{print}} &:= \{\langle -1, -1 \rangle, \langle 0, 1 \rangle, \langle 1, 1 \rangle\}. \end{aligned}$$

- The modal formulas for composite programs are defined by $R_{\pi_1; \pi_2} := R_{\pi_1} \circ R_{\pi_2}$ and $R_{\varphi?} := \{\langle x, x \rangle \mid \mathfrak{M}, x \models \varphi\}$, as usual.
- The valuation function is given by $V(\text{is_init}) := \{0, 1\}$ and $V(\text{did_print}) := \{1\}$.

Show the following:

1. $\mathfrak{M}, -1 \not\models \langle \text{run}; \text{print} \rangle \text{did_print}$,
2. $\mathfrak{M}, x \models \langle \text{init}; \text{run}; \text{print} \rangle \text{did_print}$ (for all $x \in X$),
3. $\mathfrak{M}, x \not\models \langle (\neg \text{is_init})?; \text{print} \rangle \text{did_print}$ (for all $x \in X$).

Informally speaking, the model above allows one to determine whether a program composed of initialization (`init`), doing some kind of work (`run`), and printing (`print`) is initialized or has printed something.

Suppose we want to modify the logic by counting how often we have printed, i.e. we extend the set of basic propositional letters by $\{\text{did_print}_n \mid n \in \mathbb{N}\}$. Give an appropriate model for the new logic.

References

- [Bar01] L. M. Barbosa. *Components as Coalgebras*. PhD thesis, Universidade do Minho, 2001.
- [BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Number 53 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, UK, 2001.
- [BHHK03] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous time Markov chains. *IEEE Trans. Softw. Eng.*, 29(6):524 – 541, June 2003.
- [BW99] M. Barr and C. Wells. *Category Theory for Computing Science*. Les Publications CRM, Montreal, 1999.
- [CGP99] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press, Cambridge, MA, 1999.
- [Che89] B. F. Chellas. *Modal Logic*. Cambridge University Press, Cambridge, UK, 1989.
- [Dob03] E.-E. Doberkat. Pipelines: Modelling a software architecture through relations. *Acta Informatica*, 40:37 – 79, 2003.
- [Dob07a] E.-E. Doberkat. The Hennessy-Milner equivalence for continuous-times stochastic logic with mu-operator. *J. Appl. Logic*, 35:519 – 544, 2007.
- [Dob07b] E.-E. Doberkat. *Stochastic Relations. Foundations for Markov Transition Systems*. Chapman & Hall/CRC Press, Boca Raton, New York, 2007.
- [Dob09] E.-E. Doberkat. *Stochastic Coalgebraic Logic*. EATCS Monographs in Theoretical Computer Science. Springer-Verlag, Berlin, 2009.
- [Dob10] E.-E. Doberkat. A note on the coalgebraic interpretation of game logic. *Rendiconti Ist. di Mat. Univ. di Trieste*, 42:191 – 204, 2010.
- [Dob12] E.-E. Doberkat. *Haskell für Objektorientierte*. Oldenbourg-Verlag, München, 2012.
- [DS11] E.-E. Doberkat and Ch. Schubert. Coalgebraic logic over general measurable spaces - a survey. *Math. Struct. Comp. Science*, 21:175 – 234, 2011. Special issue on coalgebraic logic.
- [Fia05] J. L. Fiadeiro. *Categories for Software Engineering*. Springer-Verlag, Berlin, Heidelberg, 2005.
- [Gir81] M. Giry. A categorical approach to probability theory. In *Categorical Aspects of Topology and Analysis*, number 915 in Lect. Notes Math., pages 68 – 85, Berlin, 1981. Springer-Verlag.
- [Lan97] S. Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer-Verlag, Berlin, 1997.

-
- [Lip11] M. Lipovača. *Learn You a Haskell for Great Good!* no starch press, San Francisco, 2011.
- [Mog89] E. Moggi. An abstract view of programming languages. Lecture Notes, Stanford University, June 1989.
- [Mog91] E. Moggi. Notions of computation and monads. *Information and Computation*, 93:55 – 92, 1991.
- [OGS09] B. O’Sullivan, J. Goerzen, and D. Stewart. *Real World Haskell*. O’Reilly, Sebastopol, CA, 2009.
- [Par85] R. Parikh. The logic of games and its applications. In M. Karpinski and J. van Leeuwen, editors, *Topics in the Theory of Computation*, volume 24, pages 111–140. Elsevier, 1985.
- [Pat04] D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame J. Formal Logic*, 45(1):19 – 33, 2004.
- [PP03] M. Pauly and R. Parikh. Game logic — an overview. *Studia Logica*, 75:165 – 182, 2003.
- [Pum99] D. Pumplün. *Elemente der Kategorientheorie*. Spektrum Akademischer Verlag, Heidelberg, 1999.
- [Pum03] D. Pumplün. Positively convex modules and ordered normed linear spaces. *J. Convex Analysis*, 10(1):109 – 127, 2003.
- [Rut00] J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 249(1):3 – 80, 2000. Special issue on modern algebra and its applications.
- [Sok05] A. Sokolova. *Coalgebraic Analysis of Probabilistic Systems*. PhD thesis, Department of Computer Science, University of Eindhoven, 2005.
- [Ven07] Y. Venema. Algebras and co-algebras. In P. Blackburn, J. van Benthem, and F. Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 331–426. Elsevier, Amsterdam, 2007.

Index

- $F(\mathcal{G})$, 5
- Meas**, 7
- Prob**, 7
- P**, 5
- Rel**, 10
- Top**, 7
- $\llbracket \varphi \rrbracket_{\mathfrak{M}}$, 89
- \mathbf{K}^{op} , 10
- Set**, 4
- AbGroup**, 120
- $\mathfrak{M}, w \models \varphi$, 89
- σ -algebra
 - product, 17
 - weak-*, 8
- $w \models_{\gamma} \varphi$, 111
- $\mathcal{L}(\Phi)$, 86
- $\mathcal{L}(\mathbb{L})$, 111
- $\mathcal{L}(\tau, \Phi)$, 87
- adjoint
 - left, 53
 - right, 53
- adjunction, 53
 - counit, 58
 - unit, 58
- algebra
 - Eilenberg-Moore, 60
 - free, 61
 - morphism, 60
 - structure morphism, 60
- Angel, 88
- atomic harmony, 98
- automaton, 9
 - with output, 30
- bisimilar
 - states, 98
- bisimulation, 71, 73, 74, 98
- bisimulation equivalence, 78
- category
 - comma, 34
 - composition, 3
 - coproduct, 19
 - injection, 19
- discrete, 5
- dual, 10
- epimorphism, 12
 - split, 116
- free, 5
- monomorphism, 11
 - split, 116
- morphism, 3
- natural transformation, 34
 - component, 34
- object, 3
- product, 16
 - projection, 16
- pullback, 22
- pushout, 26
- slice, 10
- small, 5
- sum, 19
- weak pullback, 22
- choice
 - angelic, 88
 - demonic, 88
- coalgebra, 67
 - bisimilar, 70
 - carrier, 67
 - dynamics, 67
 - mediating, 70
 - morphism, 69
- cocone, 42
 - colimit, 42
- colimit, 42
- complete
 - strongly \mathbb{S} , 104
 - weakly \mathbb{S} , 104
- cone, 40
 - limit, 41
- congruence, 83
- consistent
 - Λ , 102
 - maximal Λ , 104
- coproduct, 80
- CTL*, 113
- currying, 54

- Demon, 88
- diagram
 - chasing, 15
 - commutative, 9
- Dirac measure, 48, 49
- Eilenberg-Moore algebra, 60
 - morphism, 60
- epic, 12
 - jointly, 118
- equalizer, 118
- equivalence
 - modal, 98
- exception, 43
- formula
 - globally true, 91
 - refutable, 91
 - satisfiable, 91
- frame
 - τ , 92
 - accessibility relation, 89
 - Kripke, 89
 - morphism, 96
 - neighborhood, 91
 - set of worlds, 89
- frames
 - class of, 103
- functor, 28
 - constant, 28
 - contravariant, 29
 - covariant, 29
 - endofunctor, 28
 - forgetful, 29
 - identity, 28
 - power set, 28
- Galois connection, 55
- game
 - determined, 95
 - logic, 88
- Giry monad, 50
- Godement product, 37
- graph
 - free category, 5
 - path, 5
 - undirected, 116
- invariant subset, 84
- Kleisli
 - category, 44
 - tripel, 43
- Kripke model, 89
- left adjoint, 53
- limit, 41
- Lindenbaum Lemma, 106
- logic
 - closed
 - modus ponens, 101
 - uniform substitution, 101
 - coalgebraic
 - behavioral equivalence, 112
 - bridge operators, 115
 - logical equivalence, 112
 - predicate lifting, 108
 - modal, 101
 - normal, 102
- map
 - continuous, 7
 - kernel, 12
 - measurable, 7
- modal language
 - basic, 86
 - extended, 87
 - game logic, 88
 - nabla, 87
 - PDL, 88
- modal logic
 - normal, 102
- model
 - τ , 92
 - canonical, 106
 - image finite, 99
 - Kripke, 89
 - morphism, 97
 - neighborhood, 92
- monad, 44
 - Haskell, 51
 - Manes, 61
 - multiplication, 44
 - unit, 44
- monic, 11
 - jointly, 118

monoid, 13

morphism

- automaton, 9
- bounded, 6, 70
- codomain, 4
- domain, 4
- epi/mono factorization, 13
- epimorphism, 12
- frame, 96
- isomorphism, 14
- model, 97
- monomorphism, 11

natural transformation, 34

- component, 34
- Godement product, 37
- horizontal composition, 37
- vertical composition, 37

neighborhood

- frame, 91
- model, 92
- morphism, 100

neighborhood morphism, 100

PDL, 88

positive convexity, 63

- affine map, 64
- morphism, 64

predicate lifting, 108

probability

- discrete, 31

probability functor

- continuous, 48
- continuous space, 33
- discrete, 32, 47

propositional letters, 86

pullback, 22

- preserves weak pullbacks, 76
- weak, 22

right adjoint, 53

semi lattice, 61

space

- measurable, 7
- probability, 7
- topological, 7

state

- behavioral equivalence, 112
- logical equivalence, 112
- theory, 112

substitution, 101

terminal object, 119

Theorem

- Hennessy-Milner, 99
- Manes, 45

transition system, 6

- bisimulation, 71
- labeled, 31

upper closed, 33

world

- behavioral equivalence, 112
- logical equivalence, 112
- theory, 112

Yoneda isomorphism, 38